机器学习算法实践

推荐系统的协同过滤理论及其应用

王建芳



MACHINE LEARNING

清华大学出版社

机器学习算法实践

——推荐系统的协同过滤理论及其应用

王建芳　著

清华大学出版社
北　京

内 容 简 介

个性化推荐能够根据用户的历史行为显式或者隐式地挖掘用户潜在的兴趣和需求，并为其推送个性化信息，因此受到研究者的追捧及工业界的青睐，其研究具有重大的学术价值及商业应用价值，已广泛应用于大型电子商务平台、社交平台、新闻客户端以及其他各类旅游和娱乐类网站中。

本书内容丰富，较全面地介绍了基于协同过滤的推荐系统存在的问题、解决方法和评估策略，主要内容涉及协同过滤推荐算法中的时序技术、矩阵分解技术和社交网络信任技术等知识。

本书可供从事推荐系统、人工智能、机器学习、模式识别和信息检索等领域的科研人员及研究生阅读、参考。

图书在版编目(CIP)数据

机器学习算法实践：推荐系统的协同过滤理论及其应用/王建芳著. —北京：清华大学出版社，2018

ISBN 978-7-302-50783-3

Ⅰ. ①机…　Ⅱ. ①王…　Ⅲ. ①机器学习－算法　Ⅳ. ①TP181

中国版本图书馆 CIP 数据核字(2018)第 178631 号

责任编辑：曾　珊
封面设计：常雪影
责任校对：焦丽丽
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：三河市国英印务有限公司
经　　销：全国新华书店
开　　本：170mm×240mm　　**印　　张**：13　　**字　　数**：270 千字
版　　次：2018 年 11 月第 1 版　　**印　　次**：2018 年 11 月第1 次印刷
定　　价：69.00 元

产品编号：080054-01

前言

PREFACE

个性化推荐与信息检索技术的目标一致，也是一种帮助用户更快速地发现有用信息的工具，但与信息检索技术不同的是，个性化推荐能够根据用户的历史行为显式或者隐式地挖掘用户潜在的兴趣和需求，为其推送感兴趣并且个性化的信息，已越来越受到研究者的追捧及工业界的青睐，其研究具有重大的学术价值及商业应用价值。如今基于个性化推荐算法的推荐系统已广泛应用于大型电子商务平台（如天猫、京东和亚马逊等）、社交平台（如新浪微博、Facebook 和 Twitter 等）、新闻客户端（今日头条、天天快报等）以及其他各类旅游和娱乐类网站（如携程网、电影音乐社区等）中，在提高用户满意度和忠诚度的同时也为自身带来了可观的经济效益。

协同过滤推荐算法是个性化推荐中运用最早和最成功的一种推荐技术，它的任务是利用用户与项目评分矩阵中的已知元素来预测未知元素的评分值并将预测评分高的项目推荐给用户。协同过滤的最大优点是对推荐对象没有特殊的要求，能处理非结构化的复杂对象（如音乐、图书、电影和资讯类新闻内容等，这类产品是难以进行机器自动内容分析的信息），避免了内容分析的不完全和不精确，而且能够根据用户的历史行为推荐个性化的信息。传统的基于邻域模型的推荐算法分为数据收集（输入）、获得最近邻集合（主要是计算相似度）和预测并推荐（输出）等步骤。目前协同过滤推荐算法还存在数据的高维稀疏性、冷启动和大数据环境下扩展性等制约其进一步发展的瓶颈问题，如何解决以上问题进而提高推荐系统的推荐质量成为个性化推荐的关键，近年来基于协同过滤的推荐算法及其相关改进模型得到了学者们的广泛关注和研究。

本书作者一直从事推荐系统理论及其应用的研究工作，提出了一系列改进推荐质量的方法，并成功应用于多种复杂的实际问题。作者的这些工作大大丰富了推荐系统理论，尤其是所关注的协同过滤推荐算法对其在其他领域的进一步研究与应用奠定了技术基础，具有重要的理论意义和实际应用价值。

本书由河南理工大学计算机科学与技术学院王建芳独立完成，是作者在本领域所发表学术论文的基础上进一步加工、深化而成的，是对已有研究成果的全面总结。全书共分 5 篇 14 章。第一篇包括第 1 章，讨论了推荐算法的分类、各类算法的基本思想和改进策略，阐述推荐算法存在的问题、实验方法和评测指标。第二篇

包括第 2 章和第 3 章，主题是围绕基于时序的协同过滤推荐算法展开研究。在推荐系统中随着时间的推移，用户的关注点在不断变化，如何捕获这一动态的时间效应是个难题。本篇针对基于时序的协同过滤推荐算法展开研究。第三篇包括第 4～11 章，主题是围绕基于矩阵分解的协同过滤推荐算法展开研究。矩阵分解模型能够基于用户的行为对用户和项目进行自动分析，也就是把用户和项目划分到不同主题，这些主题可以理解为用户的兴趣和项目属性。本篇针对 SVD、概率矩阵分解、非负矩阵分解及其与相关算法的整合分别提出相关的理论。第四篇包括第 12 章和第 13 章，主题是围绕协同过滤推荐算法与社交网络的信任展开研究，将用户的评分信息和用户的社交网络信息融入传统的矩阵分解中以提高推荐质量。第五篇包括第 14 章，从实际应用的角度用 Spark 实现一个基于矩阵分解的推荐原型系统。

在本书的撰写过程中，已毕业的硕士研究生张朋飞、李骁、武文琪以及在读研究生谷振鹏、刘冉东、苗艳玲等对书稿内容和相关实验提供了大量的帮助，在此向他们表示衷心的感谢。本书的出版得到河南省高等学校重点科研项目（项目编号：15A520074）和河南理工大学博士基金的支持，在此一并表示感谢。

推荐系统所涉及的算法，尤其是协同过滤推荐算法是一个快速发展、多学科交叉的新颖研究方法，其理论及应用均有大量的问题尚待进一步深入研究。由于作者知识水平和资料获取方面的限制，书中不妥之处在所难免，敬请同行专家和读者批评指正。

作　者

2018 年 5 月

目录

CONTENTS

第一篇　基础理论

第二篇　基于时序的协同过滤推荐算法

第三篇　基于矩阵分解的协同过滤推荐算法

第一篇　基础理论

推荐系统的传统定义可以理解为“采集用户历史行为信息，结合具体推荐模型帮助用户选择商品或提供建议的过程”。现阶段完整的个性化推荐模型主要由数据收集及预处理、推荐算法和产生推荐三部分组成。

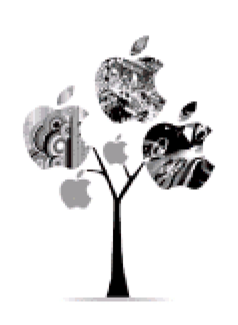
数据收集包括收集用户属性、项目属性和用户对项目的行为信息等。收集到的数据中，有些数据无法直接使用或推荐效果很差。为了后续更好地为用户提供推荐服务，需要提前对数据进行预处理——清理和减噪。

产生推荐是通过推荐算法计算得到目标用户的最近邻集合，将最近邻评价过的项目推荐给目标用户；利用模型对未知项目进行预测，将预测评分最高的项目推送给目标用户。

推荐算法作为个性化推荐系统中的核心，将收集并处理好的数据通过推荐算法为用户产生推荐。推荐算法的优劣与个性化推荐系统的推荐质量有着直接关系。

第 1 章

理 论 入 门



1.1 引言

信息技术的迅猛发展使人类社会由信息匮乏时代进入信息过载时代，而信息过载为用户在选择最中意的产品时带来沉重的处理负担。以电子商务网站为例，用户往往囿于潜在需求而无法用关键字表达或者搜索关键字表达不准确，从而不得不从浩如烟海的信息海洋获取真正需求的信息。

针对上述问题，为满足用户和企业的共同需求，满足不同用户偏好的推荐系统应运而生。此外，社会经济的快速发展带来种类繁多的产品类型，使得用户的购买目的更多地体现出固有的个体特性，在满足物质需求的基础上，推荐系统根据用户的历史行为，例如点击、购买和收藏等去挖掘用户的偏好信息，进而进行个性化推荐。为增加用户的黏性，越来越多的网站和社区开始采用推荐系统为用户提供个性化的优质服务。同时，随着 Web 3.0 时代的到来以及“互联网＋”理念的提出，人们越来越意识到推荐系统的重要性并纷纷投入其中。例如，亚马逊、eBay、天猫、京东等电子商务网站、Facebook、Twitter 和新浪微博等社交媒体均纷纷在原有业务的基础上增加推荐功能。事实表明，推荐系统的融入显著提高了用户的满意度和对网站的黏性，进而为其自身带来了可观的经济效益和社会影响力。

不过，单纯地以用户和项目为驱动的推荐引擎并不能满足相关用户的实际需要，用户在实际购买之中往往会结合自己的实际需要以及相关朋友（本书称为社交网络信息）的推荐来做最终选择，同时传统推荐算法往往带有很严重的“马太效应”。也就是说，推荐的商品往往都是热门的商品，因此造成热门的商品更加热门，而处在“长尾分布”上的商品仍得不到重视。为此，将社交网络与个性化推荐相结合提高推荐的精确度是近年来的研究热点。

在海量数据的虚拟环境下，电影网站提供的节目信息非常多，例如按演员来说，每天都会更新该演员出演的电影，包括蓝光、高清、标清和流畅等，这样每天网站上的数据量都有成千上万太字节（1TB＝1024GB），而仅仅通过一台微型计算机

或手机屏幕，希望观众找到一个自己真正喜欢的电影是不可能的。因此，社区或网站提供了一些智能导购的需要。例如京东的JIMI，根据用户的兴趣推荐用户可能感兴趣的物品，用户可以很容易地找到他们所需要的或感兴趣但不容易得到的明确的项目。而且，从实际情况来看，用户的需求往往是对商品或事件的兴趣，但目前还不清楚什么商品可以满足其潜在需求。这时，如果商家基于用户的历史行为分析出其感兴趣的信息并将这些信息呈现到用户面前，就可以把用户的潜在需求转化为现实的需求，从而给用户带来惊喜。

1.2 推荐系统的形式化定义

目前推荐系统常采用的方法主要有基于内容的推荐、基于网格的推荐、基于上下文情景的推荐和协同过滤推荐。协同过滤(Collaborative Filtering，CF)推荐技术是推荐系统中最为常用且有效的方法，可分为基于内存的协同过滤和基于模型的协同过滤，前者根据用户或者项目的相似度选出与目标用户最相似的若干用户的评分来对未评分的项目进行评分预测；后者通过分析用户和项目的内部规律，预测用户对项目的偏好，其中概率矩阵分解技术是其典型代表。目前概率矩阵分解技术还存在数据的高维稀疏性和海量数据环境下的扩展性等制约其进一步发展的瓶颈问题。如何解决以上问题进而提高推荐系统的推荐质量成为个性化推荐的关键。

一个典型的电影推荐系统一般包括含有 N 个用户的用户集合 $U=\{u_1,u_2,u_3,\cdots,u_N\}$ 和含有 M 个项目的项目集合 $I=\{i_1,i_2,i_3,\cdots,i_M\}$，每个用户 $u_i\in U$ 评价了 I 中的一部分项目，评价过的项目用 $I_{u_i}\subseteq I$ 表示，用户的打分记录往往表示成 R_{NM}，如式(1-1)所示。

$$R_{NM}=\begin{bmatrix} r_{11} & \cdots & r_{1k} & \cdots & r_{1M} \\ \vdots & & \vdots & & \vdots \\ r_{21} & \cdots & r_{2k} & \cdots & r_{2M} \\ \vdots & & \vdots & & \vdots \\ r_{N1} & \cdots & r_{Nk} & \cdots & r_{NM} \end{bmatrix} \tag{1-1}$$

式中，矩阵[①]中每一行 r_i——用户 i 评价过的电影集合，所有用户集合用 U 表示；

每一列 r_j——评价电影 j 的用户集合，所有电影集合用 V 表示；

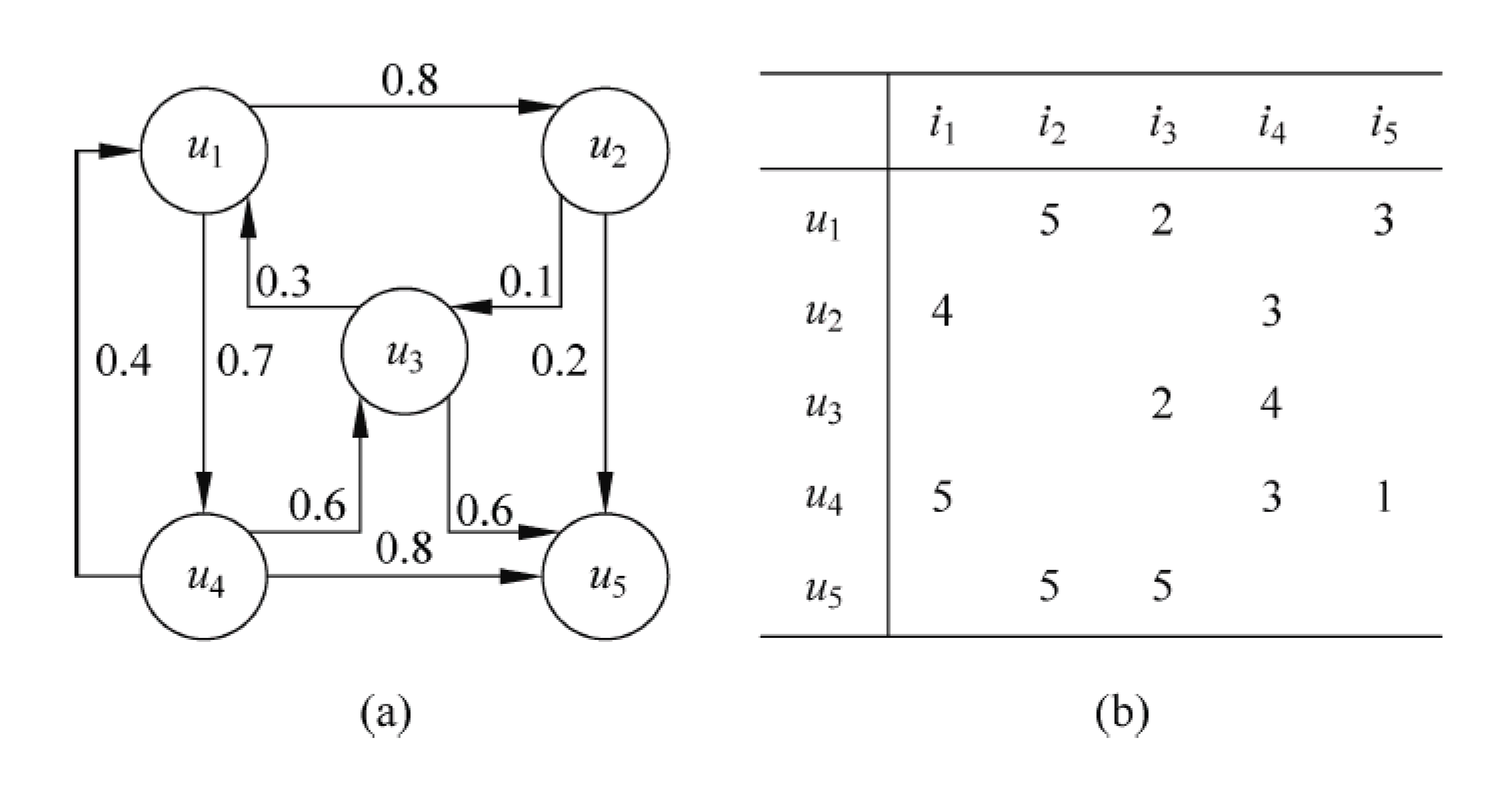
每一个元素 r_{ij}——用户 i 对电影 j 的评分，通常 r_{ij} 的取值为1～5的整数，数据越大表示用户对该项目越满意。

实际中 R_{NM} 非常稀疏，例如Ciao数据集中已有的评分数目所占比例不足1%，因此传统推荐算法的质量才会特别差。

① 注：本书中的矩阵、向量用斜体表示，而不用黑体表示。全书统一。

在现实世界中，以商品购买为例，用户的购买意图受两方面的影响，即用户本身的需要和用户朋友的推荐程度。如图 1-1 所示为基于社交网络的推荐机制示例，图 1-1(a)是用户的信任网络图，该图是一个有向图，图中包含 5 个节点(用户数)，9 条边(用户信任关系数)，每个节点代表一个用户，如果节点 i 到节点 j 存在边，则表示用户 u_i 信任用户 u_j，对应的权重表示信任程度的大小。注意，用户间的信任关系是非对称的。例如，用户 u_1 信任 u_2，但是 u_2 对 u_1 并没有明显的信任关系，不过不能主观认为 u_2 不信任 u_1，因为从图中可以看出 u_2 信任 u_3，u_3 信任 u_1，根据“六度空间”理论，u_2 对 u_1 也是具有一定的信任度的，若采用加法模型，则信任度为 0.4；若采用乘法模型，则信任度为 0.03。也就是说，信任关系是具有传递性的，同时传递算子的选择对信任度也有很大影响。

图 1-1(b)是对应的用户项目评分矩阵，矩阵中已有的值表示用户对项目的评分，缺失部分是需要预测的。以看电影为例，假设用户 u_1 想看电影 i_4，但是该用户对该电影一无所知，那么其就会求助于所信任的朋友 u_2 和 u_4，u_2 对该电影的评分是 3 分，u_4 的评分是 5 分，那么该电影很可能会吸引到用户 u_1，也就是 u_1 对 i_4 的评分也可能很高。值得注意的是，目标用户对不同用户的信任程度是不一样的。系统的目标就是利用评分矩阵和信任程度的大小精准有效地预测缺失评分并按照预测评分的高低进行推荐。



	i_1	i_2	i_3	i_4	i_5
u_1		5	2		3
u_2	4			3	
u_3			2	4	
u_4	5			3	1
u_5		5	5		

图 1-1　基于社交网络的推荐机制示例

综上所述，推荐算法的形式化定义如式(1-2)所示。

$$\forall p \in P,\quad \forall q \in Q,\quad q'_p = \arg\max_{q \in Q} u(p,q) \tag{1-2}$$

式中，P——用户集合；

Q——能够推荐给用户的物品集合；

u——一个用来计算用户 p 对物品 q 偏好程度的效用函数，计算过程可以表示为 $u: P \times Q \to R$，其中 R 为排序后的项目集合。

算法的目标是对于每个用户 p 都找到能够最大化效用函数 u 的物品子集 $Qq'_p \in Q$。

1.3 基于近邻的协同过滤推荐算法

基于近邻的协同过滤推荐算法是一种非常流行的建立推荐系统的方式，仅仅通过收集相似用户的行为而不需要用户的人口统计学信息即可自动为目标用户进行推荐。由于简单易用，协同过滤在工业界得到了飞速发展，其推荐精度主要在于相似度的选择。下面介绍基于近邻的协同过滤中常用的相似度度量方法。

1.3.1 余弦相似度

余弦相似度定义向量 a 和向量 b 为 R_{NM} 中的第 u 行和第 v 行，将两个向量的夹角余弦值定义为用户 u 和用户 v 的相似度，如式(1-3)所示。

$$\mathrm{sim}(u,v) = \cos(a,b) = \frac{a \cdot b}{|a||b|} \tag{1-3}$$

$\mathrm{sim}(u,v)$的值越接近 1，说明用户 u 与用户 v 的相似度越高。

1.3.2 修正余弦相似度

修正的余弦相似度鉴于传统的余弦相似度考虑了用户的评分偏好。也就是说，有的用户倾向于评高分，有的用户倾向于评低分。例如，两个用户对电影《西游降魔篇》和《西游伏妖篇》分别评分为 5、4 和 3、2，如果按照传统的余弦相似度来计算，那么这两个用户的相似度很低，其实这两个用户的偏好是一致的，即相对于电影《西游伏妖篇》，两个用户都更喜欢《西游降魔篇》。那么式(1-4)是用户 u 与用户 v 的修正余弦相似度。

$$\mathrm{sim}(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) \times (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_v} (r_{vi} - \bar{r}_v)^2}} \tag{1-4}$$

1.3.3 Pearson 相似度

Pearson 相似度和修正余弦相似度不同的是分母为用户的共同评分项目。式(1-5)是 Pearson 相似度。

$$\mathrm{sim}(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) \times (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{1-5}$$

1.3.4 Jaccard 相似度

Jaccard 相似度的分子为用户评分项目的交集，分母为并集，使用该相似度能大致度量用户之间的相似度。式(1-6)是 Jaccard 相似度。

$$\mathrm{sim}(u,v)=\frac{|I_u \cap I_v|}{|I_u \cup I_v|} \tag{1-6}$$

以基于用户的协同过滤为例，给定目标用户 u，可以找出与其最相似的前 n 个用户记为邻居集 N_u。对于用户 u 未进行评分的项目 i 可按照式(1-7)进行预测。

$$P_{ui}=\bar{r}_u+\frac{\sum\limits_{(v\in N_u)\wedge(r_{vi}!=0)}\mathrm{sim}(u,v)\times(r_{vi}-\bar{r}_v)}{\sum\limits_{(v\in N_u)\wedge(r_{vi}!=0)}\mathrm{sim}(u,v)} \tag{1-7}$$

由式(1-7)可对任一用户未进行评分的项目进行评分预测，将所有预测结果降序排列，从中选出前 n 个推荐给用户。

1.4 基于用户兴趣的推荐算法

近年来的大量研究表明，用户的行为记录与用户的兴趣有着极大的关联性，在推荐系统中，用户的行为时间作为一个很重要的上下文信息，用来表征用户的兴趣变化。

2000 年 Schwab 等人利用逐步遗忘的思想设计一个递减的幂函数来描述用户兴趣的变化规律，即用户最近的行为对当前兴趣的影响最大，反之用户最远的行为对当前兴趣的影响最小，并利用内容分析和协同过滤来跟踪用户兴趣。2010 年 Koren 等人提出了一种动态协同过滤推荐算法，在矩阵分解的基础上加入时间序列，动态跟踪用户兴趣变化。2010 年 Xiong 等人将时间维度作为约束项进行张量分解，利用贝叶斯定理、公式、算法进行参数自动调整，以此来改善推荐质量。2011 年 Li 等人对用户兴趣随时间变化的规律进行建模，提出一种跨时域的协同过滤推荐算法，即当用户组内用户之间的关系发生漂移时跨时域共享评分矩阵，可以追踪用户兴趣。2012 年 Ren 等人提出用户偏好模型以此来捕获用户兴趣，使用期望最大化构建代表用户偏好风格和时间动态的子空间，同时细化全局和个人偏好，以此进行迭代。2013 年 Liu 等人通过研究人们在社交网络中的在线活动和流动模式，利用矩阵分解预测用户对类别的偏好，推荐偏爱类别的相应位置来提升用户体验。2015 年 Gasmi 等人利用每个项目的类别信息，随着时间的推移反映用户喜好的动态变化，该算法利用权重函数赋予每个评分一个权重，增强用户最近和长期的兴趣信息，削弱用户短期的兴趣信息。2017 年 Yannis 等人在基于位置的社交网络中加入时间、空间和文本因素，并考虑时间维度和时间间隔对用户兴趣的影响。

2002 年赵亮等人先对评分矩阵规范化再进行 SVD 分解，对评分矩阵进行降维，利用向量空间方法得到近邻集合进行 Top-N 推荐。2007 年郑先荣等人利用遗忘曲线提出一个遗忘函数，赋予每个评分一个时间权重，以此来表示用户兴趣与最近行为关系最大。2009 年杨怀珍等人提出基于时间加权的个性化推荐算法，利用

Logistic 函数来表征用户兴趣变化规律，在计算相似度时加入时间权重。2012 年韩忠明等人为了揭示在线内容的时间动态性，利用相似度方法计算时间序列聚类问题，首先利用 Haar 小波对时间序列进行降维，然后进行增量聚类，将发展趋势相同的时间序列聚为一类，以此来产生项目推荐集。2013 年孙光福等人在概率矩阵分解的基础上对用户间的时序行为进行建模，利用用户对项目的评分时间发现用户之间的隐式关系，可以找到目标对象的最近邻集合并产生推荐，并在豆瓣数据集上验证了其算法的有效性。2015 年孙光明等人根据遗忘规律作为用户兴趣变化的度量方法，利用在一定时间段内用户对关键词的访问次数建立自适应动态兴趣度权重函数，使得推荐的项目与用户偏好一致。2017 年张应辉等人利用用户浏览项目的时长来衡量其兴趣度，时间越长其兴趣度越高，对于项目的显式属性进行分类，对于项目隐式属性可采用朴素贝叶斯算法来分析，引入参数综合考虑这些因素，最终找到最近邻集合并产生推荐。

1.5 基于模型的协同过滤推荐算法

基于模型的协同过滤一般分为聚类模型、分类模型和矩阵分解模型等。本书主要研究其中的矩阵分解模型及其与社交网络的信任相结合的算法。

1.5.1 矩阵分解模型

推荐系统中的隐语义模型，它和 Topic Model 潜在的影响因素一样，最初是在文本挖掘领域中提出来的。例如，在推荐系统中它能够基于用户的行为对用户和项目进行自动分析，也就是把用户和项目划分到不同主题，这些主题可以理解为用户的兴趣和项目属性，其中的典型代表是矩阵分解模型。

传统的矩阵分解模型有奇异值分解(Sigular Value Decomposition，SVD)、概率矩阵分解(Probabilistic Matrix Factorization，PMF)和非负矩阵分解(Non-negative Matrix Factorization，NMF)等。文献[8]提出了概率矩阵分解模型，该模型从概率生成过程角度描述矩阵分解过程，有效缓解了数据稀疏性问题；文献[9]提出了基于邻域相似度的矩阵分解模型，该模型考虑了用户兴趣相似度，进一步挖掘评分矩阵的有效信息，提高推荐精度。近年来，随着使用社交平台和社交网络的用户数量增多，依赖性增强，社交信息为协同过滤推荐算法带来了新的数据源，如好友推荐和信任用户推荐均有效促进了用户的消费行为，因为在现实生活中相对于品牌、价格和销量等参考标准，用户更倾向于信任好友的推荐。文献[11]通过调查得出相对于系统给出的推荐，用户更喜欢来自友人的推荐；文献[12]则表明大部分网站通过邀请用户和粉丝来作决策。鉴于此，大量研究人员开始运用社交信息来改进推荐算法。

概率矩阵分解模型在传统矩阵分解的基础上引入了概率的思想，假设用户和

项目的隐语义空间都服从高斯分布,同时预测评分和真实评分的误差也服从高斯分布,这样用户对项目的预测评分就是一系列的概率组合问题,然后根据最大似然估计最大化后验概率。

2008 年 Salakhutdinov 和 Mnih 等人提出概率矩阵分解算法,该算法从概率的角度来预测用户的评分,假设用户潜在因子矩阵和项目潜在因子矩阵均服从均值为 0 的球形高斯先验分布,在此假设的基础上,结合概率论和矩阵论的相关理论来预测用户对项目的偏好。2013 年,涂丹丹等将 Ma 等提出的联合概率矩阵分解并被应用到计算广告,该算法的主要思想是在用户的上下文环境约束下对用户项目评分矩阵进行分解。2015 年刁海伦融合用户项目评分矩阵信息和社交网络中显式的信任关系提出一种改进的联合概率矩阵分解模型,同时结合隐式的社交网络关系做进一步的研究,实验结果表明改进的算法在数据稀疏情况下推荐精度较高,而且对于用户冷启动问题也有一定的缓解作用。2016 年 Hernando 等人将非负矩阵分解模型应用在传统的贝叶斯概率矩阵分解模型之中,使得算法具有良好的可解释性。

概率矩阵分解算法通过优化预先设定的目标函数从而得到近似全局最优解,推荐精度较高,同时具有坚实的理论基础,能较好地应用于实践之中。但是海量数据情况下,由于用户和购买项目数量之间的关系服从幂律分布,用户少,项目多,造成数据集极度稀疏,而且由于算法不能充分挖掘用户与项目之间的关系,导致推荐精度急剧下降。

矩阵分解模型假设用户对项目的评分受到若干潜在因子的影响,将用户和项目映射到一个共同的潜在因子空间。和 Topic Model 不同的是,该类算法到底受哪种因素的影响却很难确定,正是囿于此种缺陷,一般又将矩阵分解模型称为隐语义模型,该模型最早由 Koren 于 2009 年提出。

传统的矩阵分解模型往往将固有的用户项目评分矩阵 R_{NM} 分解为两个低秩矩阵的乘积,以达到对矩阵中缺失值的预测目的,如式(1-8)所示:

$$R_{NM} \approx U_{kN}^{\mathrm{T}} V_{kM} \tag{1-8}$$

其中,$k \ll \min(M,N)$,指的是潜在因子的数量;U_{kM} 和 V_{kN} 为由分解得到的两个低秩矩阵,可以看作是用户潜在因子矩阵和项目潜在因子矩阵,往往通过迭代训练来使得 U_{kM} 和 V_{kN} 的内积不断逼近原始的用户项目评分矩阵,同时得到 U_{kM} 和 V_{kN} 后还可以对用户没有评分的项目进行评分预测。

基于矩阵分解的算法是一种学习型算法,实际中往往采用随机梯度下降(Stochastic Gradient Descent,SGD)来优化预先设定的目标函数从而得到全局最优解,而且由于潜在因子的数量 $k \ll \min(M,N)$,算法的离线计算的空间复杂度低,这在当今大数据的环境下具有很强的实用价值;同时,由于该算法有一个全局的目标函数,使得算法的预测准确率高。

1.5.2 交替最小二乘

理论研究表明交替最小二乘(Alternating Least Squares,ALS)随着迭代的进行误差会逐步降低直至收敛,ALS 完全不能保证将会收敛至全局最优解,而且在实际应用中,ALS 对初始点选取较为敏感,不恰当的选择会导致数据振荡地收敛到局部最优解。

首先按高斯分布初始化用户和项目的潜在因子向量 U 和 V。

然后固定 V,将损失函数对 V 求偏导,并令导数等于 0,得到新的用户潜在因子向量 U,如式(1-9)所示:

$$U \leftarrow (V^{\mathrm{T}}V + \lambda I)^{-1}V^{\mathrm{T}}R \tag{1-9}$$

其次固定 U,将损失函数对 U 求偏导,并令导数等于 0,如式(1-10)所示:

$$V \leftarrow (U^{\mathrm{T}}U + \lambda I)^{-1}U^{\mathrm{T}}R \tag{1-10}$$

式中,λ——正则化系数,需要实验确定。

最后便可利用得到的用户项目潜在因子空间 U 和 V 进行评分预测。

1.5.3 概率矩阵分解

概率矩阵分解是矩阵分解模型中的典型代表。图 1-2 是概率矩阵分解的概率图模型。

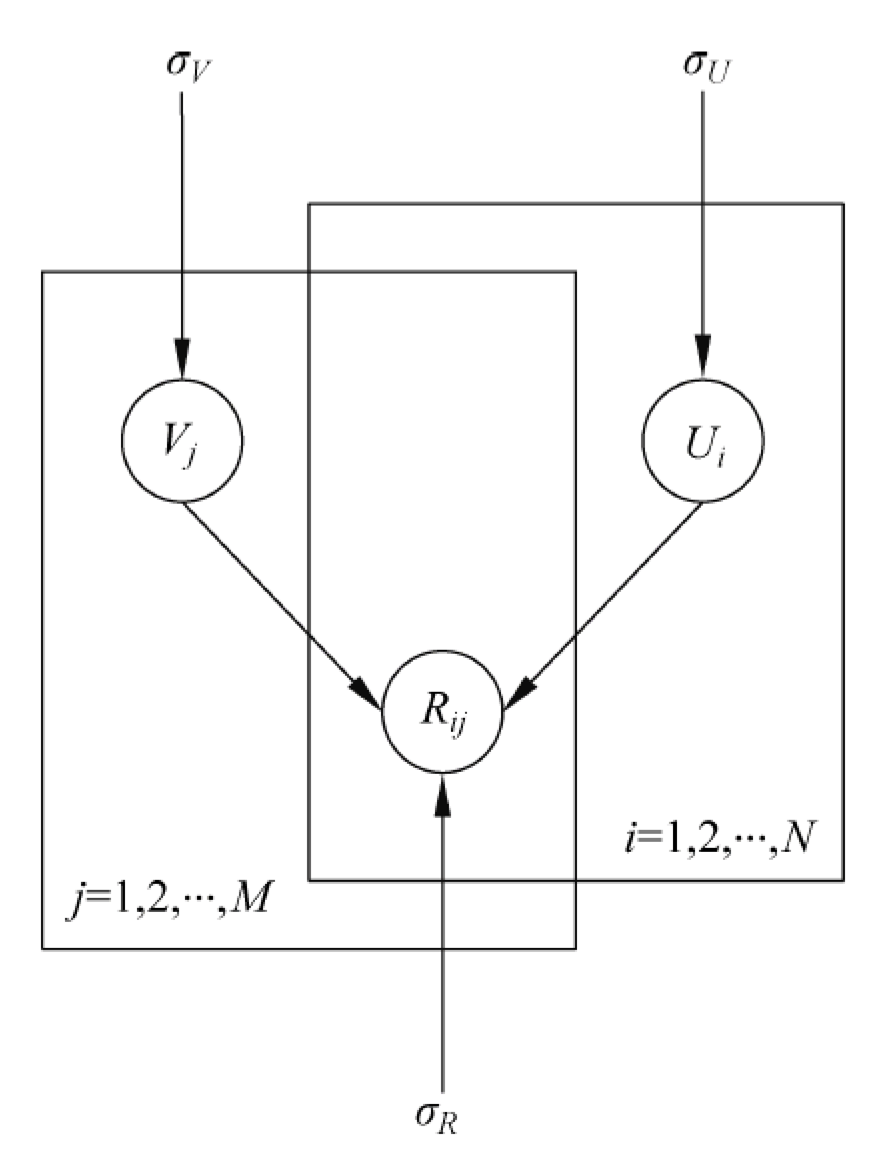


图 1-2 概率矩阵分解的概率图模型

概率矩阵分解的基本思想是在矩阵分解的基础上引入概率的思想,假设用户和商品的特征向量矩阵都符合高斯分布,如式(1-11)所示:

$$\begin{cases} p(U \mid \sigma_U^2) = \prod_{i=1}^{N} N(U_i \mid 0, \sigma_U^2 I) \\ p(V \mid \sigma_V^2) = \prod_{j=1}^{M} N(V_j \mid 0, \sigma_V^2 I) \end{cases} \tag{1-11}$$

根据上述考量，可以结合概率论和矩阵分解的相关知识预测用户对项目的喜好程度，如式(1-12)所示。

$$p(R \mid U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[N(R_{ij} \mid U_i^{\mathrm{T}}V_j,\sigma^2)]^{I_{ij}} \tag{1-12}$$

式中，$N(x|\mu,\sigma^2)$——期望为 μ、方差为 σ 的高斯分布；

I——一个指示矩阵，当且仅当 $I_{ij}=1$ 表示用户 i 选择了项目 j，否则为 0。

利用贝叶斯推导，可得用户和物品隐式特征的后验概率，如式(1-13)所示。

$$p(U,V \mid R,\sigma^2,\sigma_V^2,\sigma_U^2) \propto p(R \mid U,V,\sigma^2)\times p(U \mid \sigma_U^2)\times p(V \mid \sigma_V^2) \tag{1-13}$$

对上述预测公式取对数，如式(1-14)所示。

$$\begin{aligned}\ln p(U,V \mid R,\sigma^2,\sigma_V^2,\sigma_U^2) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\ (R_{ij}-U_i^{\mathrm{T}}V_j)^2-\frac{1}{2\sigma_U^2}\sum_{i=1}^{N}U_i^{\mathrm{T}}U_i- \\ &\frac{1}{2\sigma_V^2}\sum_{j=1}^{M}V_j^{\mathrm{T}}V_j-\frac{1}{2}\left[\left(\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\right)\ln\sigma^2+\right. \\ &\left.ND\ln\sigma_U^2+MD\ln\sigma_V^2\right]+C\end{aligned} \tag{1-14}$$

式中，C——一个不依赖于模型超参数的常量。

最大化 U 和 V 的后验概率等于最小化式(1-15)。

$$\arg\min_{U,V}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\ (R_{ij}-U_i^{\mathrm{T}}V_j)^2+\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{\mathrm{Fro}}^2+\frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|_{\mathrm{Fro}}^2 \tag{1-15}$$

式中，Fro——F 范数；

λ_U 和 λ_V——正则化系数，防止过拟合。

然后利用 SGD 来训练，如式(1-16)所示。

$$\begin{cases}U \leftarrow U-\eta\times\dfrac{\partial L}{\partial U}\\ V \leftarrow V-\eta\times\dfrac{\partial L}{\partial V}\end{cases} \tag{1-16}$$

式中，η——SGD 的学习速率。

概率矩阵分解模型能较好地适应大规模数据集(与其他矩阵分解算法比较)，时间复杂度随观测数据量的增长呈线性增长；同时，实验结果表明，在数据非常稀疏的情况下有更好的预测性能。

1.5.4 非负矩阵分解

非负矩阵分解是把一个矩阵分解成两个矩阵乘积的形式，以此来分解多维数据。

对于现代化推荐系统，需要处理的数据量非常庞大，在现有矩阵分解的基础上 Lin 提出了一种时间复杂度比较低的 NMF 算法。该算法利用每一个已知评分项更新分解后的用户-隐因子矩阵 $P_{m\times k}$ 和项目-隐因子矩阵 $Q_{k\times n}$。在 Lin 算法的基础

上，为了防止分解后的矩阵出现过拟合，加入正则项的乘性迭代，如式(1-17)所示。

$$\begin{cases} p_{u,k} = p_{u,k} \cdot \dfrac{\sum\limits_{i \in I_u} q_{k,i} \cdot r_{u,i}}{|I_u|\lambda_p p_{u,k} + \sum\limits_{i \in I_u} q_{k,i} \cdot r^*_{u,i}} \\ q_{k,i} = q_{k,i} \cdot \dfrac{\sum\limits_{u \in Ui} p_{u,k} \cdot r_{u,i}}{|U_i|\lambda_q q_{k,i} + \sum\limits_{i \in I_u} p_{u,k} \cdot r^*_{u,i}} \end{cases} \tag{1-17}$$

式中：I_u——评分不为零的项目集合；

U_i——评分不为零的用户集合；

$r_{u,i}$——用户 u 对项目 i 的实际评分；

$r^*_{u,i}$——预测的用户 u 对 i 的评分，其可以由初始化的用户-隐因子矩阵 $P_{m\times k}$ 和项目-隐因子矩阵 $Q_{k\times n}$ 计算得到。

1.6 基于信任的协同过滤推荐算法

针对矩阵分解算法在海量数据情况下推荐精度急剧下降的问题，国内外学者提出融合其他信息来约束传统的概率矩阵分解，其中信任信息是一种既容易获得又十分有效的信息。

推荐系统中对信任信息的研究主要集中在信任的度量方式，也就是信任的传播与聚合上，传统的信任度量方法有 TidalTrust 和 MoleTrust 等。TidalTrust 是一个递归算法，在 Golbeck 等人的实验中不是考虑用户 a 到用户 c 的直接信任关系，而是首先找到 a 到 c 的其他全部路径，分别计算信任值，据此得到如下结论：①越短的路径产生的信任值越准确；②包含越高信任值的路径产生的信任结果比较好。结合结论①和六度空间理论，作者使用广度优先搜索算法来计算最短路径上前 6 个节点的信任值。另外，实际情况中信任的计算也和商品的评分一样因人而异。也就是说，有的人倾向于给出高的信任值，有的人倾向于给出较低的信任值，即便很信任对方。因此，需要首先计算出链接路径上每个用户的所有路径中信任值的最大值，以该值作为最小的信任阈值做加权处理，以此得到局部信任值。值得注意的是，作者在文末提出使用 TidalTrust 算法不一定比使用信任平均值的协同过滤推荐算法效果更好，仅当某些用户的偏好显著偏离平均值时效果才更好。

MoleTrust 由 Massa 等人提出，首先去掉数据集中的环得到有向无环图，这样每个用户不会重复计算，提高了算法的实际效率；其次计算和目标用户距离在 2 度之内节点之间的信任值，2 度之外节点之间信任值的计算方式和 TidalTrust 类似。另外，与 TidalTrust 不同的是，MoleTrust 算法进行加权处理时考虑所有对物品评分而且目标用户能够达到的用户集，并且只有那些信任度超过特定阈值的用户的信任度才被考虑在内，该阈值由用户自己指定。MoleTrust 计算的也是局部信任值，该算法比类似 PageRank 的全部信任度算法具有更好的预测效果，尤其是

对那些比较具有争议的用户（这些用户被一些人信任而被另外一些人不信任），同时该算法对冷启动用户的预测效果更好。表 1-1 列出了 TidalTrust 和 MoleTrust 算法的异同。

表 1-1　TidalTrust 和 MoleTrust 算法的异同

算　　法	TidalTrust	MoleTrust
传播	乘法	乘法
聚合	基于信任的加权平均	基于信任的加权平均
路径最大值	动态	静态
信任阈值	动态	静态
传播中的 TTP 需求	最短路径	最短路径且小于阈值
评分预测	基于信任的加权平均	基于信任的协同过滤

这些经典的信任度量方法虽然能作为互联网中信任度的度量，但是本身也存在一些缺陷，例如上述算法不能度量本身不存在直接联系用户之间的信任程度。

鉴于此，2013 年西安交通大学秦继伟博士将用户的情感需求作为调节背景，融合社交网络中的信任机制和用户的情感偏好提出一种改进的推荐算法，在推荐系统领域公开的数据集上的实验结果表明，改进算法对于数据稀疏性和虚假评分具有一定的缓解作用；同年，王海艳等人提出结合用户自身的特征来改进传统的基于协同过滤的服务选择模型，同时结合层次分析法确定各个属性的权重值，仿真实验表明该算法不仅提高了推荐的质量，同时对于攻击具有一定的健壮性。2014 年 Zeng 等人提出一种社交网络中的混合信任聚合模型，信任值的计算过程中不仅考虑用户间的直接信任关系和间接信任关系，也同时考虑一个用户如何被组内的其他用户所信任，实验结果表明此信任计算方式能合理地计算出所有用户之间的信任值；同年，朱强等人结合社会网络分析中的凝聚子群提出一种改进的协同过滤推荐算法，在各个子群中进行朋友推荐和服务计算。2015 年 Ma 等人提出一种融合标签信息的扩展协同过滤方法，在两个公开数据集上的实验结果表明，新算法在提升推荐精度的前提下对冷启动项目的推荐精度也有显著提升；同年，Shen 等人提出一种在线社交网络的信任计算方法，增量地计算社交网络用户之间的信任值，包括基于用户的协同过滤、基于项目的协同过滤、奇异值分解和基于信任的协同过滤的实验结果表明，此种增量计算方法能显著提高用户的黏性。2016 年 Chen 等人通过一种信任矩阵分解形式来融合传统的信任传播和信任聚合计算方式，在多个数据集上的实验结果表明，该方法能显著提高推荐精度，尤其是在冷启动情况下相对于传统的其他方法提升效果更明显；同年，Gohari 等人提出一种信任机制增强协同过滤推荐算法的推荐模型，结合具有共同评分用户之间的稀疏信任关系提出一种新的信任传播算法，实验结果表明该算法对稀疏用户的推荐精度很高，而且对冷启动用户效果更好；同年，Meyffret 等人利用局部信任关系进行个性化推荐，更进一步提升了推荐的准确度。

1.7 推荐系统现存问题

矩阵分解应用在推荐系统中目前存在四类问题，即冷启动问题、数据稀疏性问题、可扩展性问题和易受攻击性问题。冷启动主要是为了解决新用户和新项目的推荐问题，易受攻击主要是为了缓解用户的恶意评分，从而提升相关用户的知名度或者提升相关项目被推荐的次数。

1.7.1 冷启动

在推荐系统中，冷启动问题主要表现在以下几方面：当新用户加入系统时，没有浏览或评价过任何产品，没有用户的行为数据，所以也就无法根据用户的历史行为预测其兴趣，从而无法为新用户提供推荐服务；当系统加入新项目时，没有用户对其评价过，也不能被推荐；在一个新开发的个性化推荐系统中，如何在系统一发布就可以让用户体验到个性化推荐服务。

Bedi 等人利用 Facebook 社交网络上用户之间的互动，试图处理冷启动问题。Facebook 是一个很受欢迎的社交网站，朋友或熟人的选择往往会影响用户的意见或选择，可以利用这个思想来为用户提供推荐；提出一个 IBSP 算法，利用社会交往因子克服冷启动问题；使用 Java 开发的一个图书原型系统，用 Facebook 的 API 图形从用户的社交图中提取信息。于洪等人利用用户注册信息（年龄、性别、职业、民族、居住地等）和项目内容信息（项目的详细描述）分别进行建模，提供推荐服务。Le 等人提出一种新的相似度度量方法——NHSM 来解决用户冷启动问题。

1.7.2 数据稀疏性

在传统推荐算法的研究过程中，往往具有海量的用户和项目信息。也就是说，用户和项目的潜在因子矩阵是高维稀疏的，由此导致任意两个向量之间近似正交，计算得到的相似度往往为零，传统的基于相似度计算的模型将得不到理想的结果。因此，评价数据集的稀疏度显得十分必要，实际应用中往往采用用户项目评分矩阵中未评分数据量占评分总量的比例作为稀疏度的衡量指标，稀疏度越大，传统算法的精度越低，也就越难处理。

1.7.3 可扩展性

在大数据环境下，由于用户量和数据量巨大，传统的矩阵分解算法响应缓慢，同时存储成本较高，这就限制了传统的矩阵分解算法在实际中的应用。有鉴于此，改进的算法复杂度要越低越好，同时通过分布式文件系统（Hadoop Distributed File System，HDFS）来存储数据，考虑计算效率，这时可将矩阵分解算法进行并行化操作，以此来提高算法对海量数据的处理能力。

1.7.4 用户兴趣漂移

由于用户的兴趣爱好瞬息变化，存在用户兴趣漂移问题，给推荐系统带来极大的挑战，影响推荐的实时性。引起用户兴趣漂移的主要原因：由于年龄增长或转换生活状态，用户自身的兴趣和关注点会有不同；用户兴趣受新闻事件和项目流行度的影响；用户对项目的兴趣会受到季节效应和节日的影响。例如，当用户在网上看电影时，今天因为新电影的上映或其他原因喜欢某一主题的电影，明天又会因为其他原因关注另外一种主题的电影，又或是因为有其他人的加入而观看别人喜爱的类型电影，用户的兴趣随时间、节日和人物变化而变化。

由于上述问题以及各方面的原因，都会导致推荐质量下降。所以算法改进的最终目的是向用户准确推荐项目，所推荐的结果使得用户满意。

1.8 评测指标

推荐算法的评测指标很多，针对推荐系统的侧重点不同，其评价标准也不尽相同。衡量算法的评分预测准确度时多采用平均绝对误差（Mean Absolute Error，MAE）和均方根误差（Root Mean Square Error，RMSE）指标；针对 Top-N 推荐的预测准确率时一般通过准确率（Precision）、召回率（Recall）度量 F 值（F-Measure）和 $P(u)@N$ 指标。

推荐结果的召回率定义如式(1-18)所示。

$$\text{Recall} = \frac{\sum_{u \in U} |R_u \cap T_u|}{\sum_{u \in U} |T_u|} \tag{1-18}$$

推荐结果的精确率定义式(1-19)所示。

$$\text{Precision} = \frac{\sum_{u \in U} |R_u \cap T_u|}{\sum_{u \in U} |R_u|} \tag{1-19}$$

综合考虑精确率和召回率，将两者融合在一起形成 F 指标，定义式(1-20)所示。

$$F = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Precision} + \text{Recall}} \tag{1-20}$$

Top-N 推荐列表中的项目与测试集中评分最高项目的匹配度定义如式(1-21)所示。

$$P(u)@N = \frac{\#\text{relevant items in top } N \text{ items for } u}{N} \tag{1-21}$$

为了衡量推荐系统发掘长尾的能力，使用覆盖率（Coverage）来计算系统所推荐的项目占项目集合的比例，这是商家最关心的指标，其定义如式(1-22)所示。

$$\text{Coverage} = \frac{|U_{u \in U} R_u|}{|n|} \tag{1-22}$$

推荐列表 R_u 的多样性定义式(1-23)所示。

$$\text{Diversity}(R_u) = \frac{\sum\limits_{i,j \in R_u, i \neq j} (1 - \text{sim}(i,j))}{\frac{1}{2} R_u (R_u - 1)} \tag{1-23}$$

系统整体的多样性定义式(1-24)所示。

$$\text{Diversity} = \frac{1}{m} \sum_{u \in U} \text{Diversity}(R_u) \tag{1-24}$$

本书主要采用 MAE 和 RMSE 指标衡量评分预测准确度，MAE 用来衡量推荐的精确率，能很好地反映预测值误差的实际情况。设在训练集上得到用户的预测评分集合为 $p = \{p_{u,1}, p_{u,2}, p_{u,3}, \cdots, p_{u,n}\}$，用户实际评分集合为 $r = \{r_{u,1}, r_{u,2}, r_{u,3}, \cdots, r_{u,n}\}$，通过计算两集合评分的差值来衡量推荐的精确率。MAE 定义如式(1-25)所示。

$$\text{MAE} = \frac{\sum\limits_{u,i \in N} |r_{u,i} - p_{u,i}|}{N} \tag{1-25}$$

RMSE 同样用来衡量推荐的精确率，RMSE 更侧重于预测评分与实际评分差值的绝对值，相对 MAE 加大了惩罚力度，RMSE 值越小，则 MAE 值越小，推荐精确度越高。RMSE 定义如式(1-26)所示。

$$\text{RMSE} = \sqrt{\frac{\sum\limits_{u,i \in N} |r_{u,i} - p_{u,i}|^2}{N}} \tag{1-26}$$

本章小结

本章对推荐算法的相关知识进行了介绍，讨论推荐算法的分类及各算法的基本思想，并着重分析基于内存协同过滤推荐算法，阐述推荐算法存在的问题、实验方法和评测指标，只有充分了解协同过滤推荐算法才能针对其做进一步的研究。

参考文献

[1] Guo G，Zhang J，Yorke-Smith N. TrustSVD：Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings[C]//Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI Press，2015：123-129.

[2] Fernández-Tobías I，Braunhofer M，Elahi M，et al. Alleviating the New User Problem in Collaborative Filtering by Exploiting Personality Information[J]. User Modeling and User-Adapted Interaction，2016，26(2)：221-255.

[3] 王立才,孟祥武,张玉洁.上下文感知推荐系统[J].软件学报,2012,23(1):1-20.

[4] He R, Mcauley J. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering[C]//International Conference on World Wide Web. International World Wide Web Conferences Steering Committee,2016:507-517.

[5] Silva E Q D,Camilo-Junior C G,Pascoal L M L,et al. An Evolutionary Approach for Combining Results of Recommender Systems Techniques Based on Collaborative Filtering [J]. Expert Systems with Applications,2016,53:204-218.

[6] Zhang J,Lin Y,Lin M,et al. An Effective Collaborative Filtering Algorithm Based on User Preference Clustering[J]. Applied Intelligence,2016,45(2):1-11.

[7] Yang B,Lei Y,Liu J,et al. Social Collaborative Filtering by Trust. [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2016,PP(99):2747-2753.

[8] Zheng X,Luo Y,Sun L,et al. A New Recommender System Using Context Clustering Based on Matrix Factorization Techniques[J]. Chinese Journal of Electronics, 2016, 25(2): 334-340.

[9] Kushwaha N,Sun X,Vyas O P,et al. SemPMF: Semantic Inclusion by Probabilistic Matrix Factorization for Recommender System[J]. 2016,27(2):294-310.

[10] Qiu Y, Lin C J, Juan Y C, et al. Recosystem: Recommender System using Matrix Factorization[J]. 2015,45(3):39-47.

[11] Nguyen G T, Ahn H. A Combining Method of Content-based Information into Matrix Factorization Recommendation System[J]. 2016,53:204-218.

[12] Pirasteh P, Hwang D, Jung J J. Exploiting Matrix Factorization to Asymmetric User Similarities in Recommendation Systems[J]. Knowledge-Based Systems, 2015, 83(1): 51-57.

[13] 何佳知.基于内容和协同过滤的混合算法在推荐系统中的应用研究[D].上海:东华大学,2016.

[14] 刘晓光.基于遗忘理论和加权二部图的推荐系统研究[D].贵阳:贵州大学,2015.

[15] 王立才.上下文感知推荐系统若干关键技术研究[D].北京:北京邮电大学,2012.

[16] 冷亚军,陆青,梁昌勇.协同过滤推荐技术综述[J].模式识别与人工智能,2014,27(8):720-734.

[17] 荣辉桂,火生旭,胡春华,等.基于用户相似度的协同过滤推荐算法[J].通信学报,2014(2):16-24.

[18] 朱夏,宋爱波,东方,等.云计算环境下基于协同过滤的个性化推荐机制[J].计算机研究与发展,2014,51(10):2255-2269.

[19] 洪舒怡.基于矩阵分解的推荐系统模型和算法改进研究[D].厦门:厦门大学,2016.

[20] Gomez-Uribe C A, Hunt N. The Netflix Recommender System: Algorithms, Business Value,and Innovation[J]. Acm Transactions on Management Information Systems,2016,6(4):13.

[21] Rampure V,Tiwari A. A Rough Set Based Feature Selection on KDD CUP 99 Data Set [J]. International Journal of Database Theory & Application,2015,8.

[22] 赵恒.基于LBS的本地美食推荐系统的研究与实现[D].成都:电子科技大学,2015.

[23] Matuszyk P,Vinagre J,Spiliopoulou M,et al. Forgetting Methods for Incremental Matrix Factorization in Recommender Systems[C]//ACM Symposium on Applied Computing.

ACM,2015: 947-953.
[24] Zhao C,Peng Q,Zhang Z. A Matrix Factorization Algorithm with Hybrid Implicit and Explicit Attributes for Recommender Systems[J]. Journal of Xian Jiaotong University,2016.
[25] 陈克寒,韩盼盼,吴健. 基于用户聚类的异构社交网络推荐算法[J]. 计算机学报,2013,36(2): 349-359.
[26] 金淳,张一平. 基于 Agent 的顾客行为及个性化推荐仿真模型[J]. 系统工程理论与实践,2013,33(2): 463-472.
[27] Lin H,Yang X,Wang W,et al. A Performance Weighted Collaborative Filtering Algorithm for Personalized Radiology Education [J]. Journal of Biomedical Informatics, 2014, 51: 107.
[28] 乌达巴拉,汪增福. 基于半监督的短语情感倾向性分析方法[J]. 模式识别与人工智能,2016,29(4): 289-297.
[29] 张锋,常会友. 使用 BP 神经网络缓解协同过滤推荐算法的稀疏性问题[J]. 计算机研究与发展,2015,43(4): 667-672.
[30] Boutet A, Frey D, Guerraoui R, et al. Privacy-Preserving Distributed Collaborative Filtering[J]. Computing,2016,98(8): 827-846.
[31] Liu J,Tang M,Zheng Z,et al. Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation[J]. IEEE Transactions on Services Computing,2016: 1-1.
[32] Nasi R,Taber A,Vliet N V. Empty Forests,Empty Stomachs? Bushmeat and Livelihoods in the Congo and Amazon Basins[J]. International Forestry Review,2016,13(3): 14.
[33] Sampoornam M. Collaborative Based Filtering Approach for Web Service Recommendations Using GEO Locations[J]. 2015,3(3): 1045-1047.
[34] Jr E C T, Ferrucci P, Duffy M. Facebook Use, Envy, and Depression Among college Students: Is Facebooking Depressing? [J]. Computers in Human Behavior,2015,43(43): 139-146.
[35] Kingsbury B E D, Sainath T N, Sindhwani V. Low-rank Matrix Factorization for Deep Belief Network Training with High-dimensional Output targets[J]. 2016: 6655-6659.
[36] 涂丹丹,舒承椿,余海燕. 基于联合概率矩阵分解的上下文广告推荐算法[J]. 软件学报,2013(3): 454-464.
[37] 刁海伦. 基于社交网络的个性化推荐算法研究[D]. 天津: 天津师范大学,2015.
[38] Hernando A, Lazaro J, Gil E, et al. Inclusion of Respiratory Frequency Information in Heart Rate Variability Analysis for Stress Assessment. [J]. IEEE Journal of Biomedical & Health Informatics,2016,20(4): 1016-1025.
[39] Shambour Q,Lu J. An Effective Recommender System by Unifying User and Item Trust Information for B2B Applications[J]. Journal of Computer & System Sciences,2015,81(7): 1110-1126.
[40] Zheng X L,Chen C C,Hung J L,et al. A Hybrid Trust-Based Recommender System for Online Communities of Practice[J]. IEEE Transactions on Learning Technologies,2015,8(4): 345-356.
[41] Golbeck J. Personalizing Applications through Integration of Inferred Trust Values in Semantic Web-based Social Networks [J]. Proceedings of Semantic Network Analysis Workshop,2005.

[42] Wu Z, Yu X, Sun J. An Improved Trust Metric for Trust-Aware Recommender Systems [C]//International Workshop on Education Technology and Computer Science. IEEE, 2009: 947-951.
[43] 秦继伟,郑庆华,郑德立,等. 结合评分和信任的协同推荐算法[J]. 西安交通大学学报, 2013,47(4): 100-104.
[44] 王海艳,张大印. 一种可信的基于协同过滤的服务选择模型[J]. 电子与信息学报,2013,35 (2): 349-354.
[45] Zeng J, Gao M, Wen J, et al. A Hybrid Trust Degree Model in Social Network for Recommender System [C]//Iiai, International Conference on Advanced Applied Informatics. IEEE, 2014: 37-41.
[46] 朱强,孙玉强. 一种基于信任度的协同过滤推荐方法[J]. 清华大学学报(自然科学版), 2014(3): 360-365.
[47] Ma T, Zhou J, Tang M, et al. Social Network and Tag Sources Based Augmenting Collaborative Recommender System[J]. Ieice Transactions on Information & Systems, 2015, E98. D(4): 902-910.
[48] Shen X, Long H, Ma C. Incorporating Trust Relationships in Collaborative Filtering Recommender System [C]//IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, NETWORKING and Parallel/distributed Computing. IEEE, 2015: 1-8.
[49] Parnes P, Synnes K, Schefström D. *m*-Tunnel: A Multicast Tunneling System with a User-based Quality-of-Service Model[J]. Springer, 2016, 1309: 87-96.
[50] Li D, Chen C, Lv Q, et al. An Algorithm for Efficient Privacy-Preserving Item-based Collaborative Filtering[J]. Future Generation Computer Systems, 2016, 55: 311-320.
[51] 潘骏驰,张兴明,汪欣. 融合用户可信度的改进奇异值分解推荐算法[J]. 小型微型计算机系统, 2016, 37(10): 2171-2176.
[52] Chen C, Zheng X, Zhu M, et al. Recommender System with Composite Social Trust Networks[J]. International Journal of Web Services Research, 2016, 13(2): 56-73.
[53] Gohari F S, Haghighi H, Aliee F S. A Semantic-enhanced Trust based Recommender System Using Ant Colony Optimization[J]. Applied Intelligence, 2016: 1-37.
[54] Gillis N, Vavasis S A. Fast and Robust Recursive Algorithmsfor Separable Non-negative Matrix Factorization[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 2014, 36(4): 698-714.
[55] 张志绮. 基于用户关系的矩阵分解推荐算法研究[D]. 北京: 北京交通大学, 2016.
[56] 王升升,赵海燕,陈庆奎,等. 基于社交标签和社交信任的概率矩阵分解推荐算法[J]. 小型微型计算机系统, 2016, 37(5): 921-926.
[57] Song Q, Cheng J, Lu H. Incremental Matrix Factorization via Feature Space Re-learning for Recommender System[C]//The, ACM Conference, 2015: 277-280.
[58] Salakhutdinov R, Mnih A. Probabilistic Matrix Factorization. [J]. Advances in Neural Information Processing Systems, 2015: 1257-1264.
[59] Hernando A, Bobadilla J, Ortega F. A Non-negative Matrix Factorization for Collaborative Filtering Recommender Systems Based on a Bayesian Probabilistic Model[J]. Knowledge-Based Systems, 2016, 97(C): 188-202.

[60] Lee H,Kwon J. Improvement of Matrix Factorization-based Recommender Systems Using Similar User Index[J]. International Journal of Software Engineering & Its Applications, 2015,9.
[61] Boutet A,Frey D,Guerraoui R,et al. Privacy-preserving Distributed Collaborative Filtering [J]. Computing,2016,98(8): 827-846.
[62] Yu M C,Wu Y C J,Alhalabi W,et al. Research Gate[J]. Computers in Human Behavior, 2016,55(PB): 1001-1006.
[63] 朱夏,宋爱波,东方,等. 云计算环境下基于协同过滤的个性化推荐机制[J]. 计算机研究与发展,2014,51(10): 2255-2269.
[64] 杜永萍,黄亮,何明. 融合信任计算的协同过滤推荐方法[J]. 模式识别与人工智能,2014, 27(5): 417-425.
[65] 顾梁,杨鹏,罗军舟. 一种播存网络环境下的 UCL 协同过滤推荐方法[J]. 计算机研究与发展,2015,52(2): 475-486.
[66] 刘胜宗,廖志芳,吴言凤,等. 一种融合用户评分可信度和相似度的协同过滤推荐算法[J]. 小型微型计算机系统,2014,35(5): 973-977.
[67] 张佳,林耀进,林梦雷,等. 基于目标用户近邻修正的协同过滤推荐算法[J]. 模式识别与人工智能,2015,28(9): 802-810.
[68] 杨丽,钮心忻,黄玮. 基于协同谱聚类的推荐系统托攻击防御算法[J]. 北京邮电大学学报,2015,38(6): 81-86.
[69] 李贵,王爽,李征宇,等. 基于时间加权三部图的分众分类标签推荐算法[J]. 小型微型计算机系统,2016,37(2): 269-274.
[70] 高升,任思婷,郭军. 基于潜在因子模型的跨领域信息推荐算法[J]. 电信科学,2015,31 (7): 75-79.
[71] Zeng J,Leng B,Xiong Z. 3-D Object Retrieval Using Topic Model[J]. Multimedia Tools and Applications,2015,74(18): 7859-7881.
[72] Koren Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2008: 426-434.
[73] Koren Y. Collaborative Filtering with Temporal Dynamics[J]. Communications of the ACM,2010,53(4): 89-97.
[74] Levy O,Goldberg Y. Neural Word Embedding as Implicit Matrix Factorization[J]. Advances in Neural Information Processing Systems,2014,3: 2177-2185.
[75] Lian D, Zhao C, Xie X, et al. GeoMF: Joint Geographical Modeling and Matrix Factorization for Point-of-Interest Recommendation[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2014: 831-840.

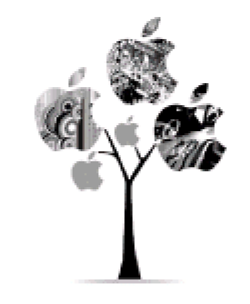
第二篇 基于时序的协同过滤推荐算法



目前协同过滤遇到很多问题和挑战,其时效性是主要问题,严重影响了推荐质量。由于传统的协同过滤算法忽略了随着时间变化而用户的兴趣也在不断发生变化这一问题,即存在用户兴趣漂移现象。传统的协同过滤推荐算法只是单一地通过评分来分析用户的兴趣爱好,统一地将评分用 1～5 分代表用户的喜爱程度,其时效性不足。用户的兴趣偏好不但范围广泛,而且实时变化,例如:一个孩子在几岁时可能对动画片感兴趣,在青春期可能对浪漫爱情片感兴趣,随后又可能对文艺片感兴趣,再过几年可能对剧情片感兴趣等。随着时间推移,用户的关注点在不断变化,如何捕获这一动态的时间效应是一个难题。

第 2 章

基于巴式系数改进相似度的协同过滤推荐算法



针对传统协同过滤推荐算法中评分数据稀疏性及所造成的推荐质量不高问题，提出一种巴氏系数(Bhattacharyya Coefficient)改进相似度的协同过滤推荐算法。在基于近邻协同过滤推荐算法基础上，利用巴氏系数改进相似度计算方法，在计算相似度时不仅依靠两个用户间的共同评分而是所有评分，首先利用 Jaccard 相似度来计算用户间的全局相似度；其次使用巴氏系数获得评分分布的整体规律，并结合 Pearson 相关系数来计算其局部相似度；最后融合全局相似度和局部相似度得到最终的相似度矩阵。在 Movielens 数据集的实验结果表明，该算法在稀疏数据集上获得更好的推荐结果，有效地缓解了评分数据稀疏性问题，提高了推荐的准确度。

2.1 引言

推荐系统帮助人们成功解决信息过载问题，并且在过去的几十年建立了电子商务的重要组成部分。推荐系统的首要任务是通过对大型项目空间的滤波为用户提供项目或产品的个性化推荐。很多推荐算法在电子商务、数字图书馆、电子媒体和在线广告等各种应用中发展，其中协同过滤是迄今为止应用最广泛、最成功的个性化推荐技术。

基于近邻的协同过滤作为协同过滤中重要的一类，因其简单、直观、有效的特点被广泛应用于电影、音乐、图书等领域。其基本思想是通过用一种相似度度量方法找到最近邻居集合(目标用户的最相似用户集合或项目的最相似项目集合)，最后通过对最近邻居集合的评分进行加权平均求和，从而产生推荐集。相似度计算的准确性直接影响推荐质量，传统的相似度度量方法利用两个用户对相同项目的评分(共同评分)来计算用户之间相似度；项目之间相似度通过对项目共同评分用户的评分计算得到。然而，当数据极大稀疏时无法保证足够的共同评分数据，传统的相似度计算方法推荐效果不佳。即使没有一个共同评分用户，两个项目也可以

是相似的；用户评价的项目不同，两个用户也可以是相似的，这些情况是传统的相似度计算方法没有考虑到的。因此，传统的相似度计算方法并不适合于稀疏数据（共同用户很少或者共同评价项目很少甚至没有）。

文献[4]在社交网络中引入了用户相似度概念，提出了基于用户相似度的协同过滤推荐算法，以改善社交网络中用户好友推荐的问题。文献[5]提出了基于用户模糊相似度的协同过滤推荐算法，分别建立年龄和评分的梯形模糊模型，将用户年龄和评分模糊化，进行相似度计算。文献[6]基于搜索引擎日志，提出了一种基于流行性和相似度相结合的查询推荐策略，为目标用户产生推荐词集合，改善推荐的多样性和流行性。文献[7]提出了一种结合 Jaccard 相似度和 Pearson 相关系数的改进相似度度量方法，在计算相似度时不但考虑共同评分还考虑共同评分的绝对数量，提供精确的评分预测。J. Bobadilla 提出了基于 MSD 的相似度度量方法，并在此基础上将评分数据结合上下文信息，提出 Jaccard 与 MSD 相结合的相似度来提高 Pearson 相关性的结果[8]。上述改进相似度计算方法在计算相似度时只考虑了相似度的局部信息，没有考虑其全局相似度。

针对上述问题，本章提出了一种利用巴氏系数改进相似度的协同过滤推荐算法（BCCF），改进相似度计算重视用户的每一个评分，在计算用户之间相似度时不单单只考虑用户评分之间的相似度。分析传统相似度方法的优缺点，结合 Jaccard 和 Pearson 方法的优点，利用巴氏系数来发现用户评分的分布规律，在 Movielens 数据集的实验表明，BCCF 算法在相对稀疏的数据集上的表现更好，可以有效改善评分数据稀疏性问题，提高预测评分的准确性。

2.2 相关工作

协同过滤是推荐系统中应用最广泛的算法之一，其关键在于最近邻居集合的选择，推荐系统的性能直接依赖于目标用户（项目）邻居的选择，也就是依赖于用户（项目）之间相似度的度量，因而改进相似度计算方法成为缓解数据稀疏性问题、提高推荐质量的有效途径之一。协同过滤中常用的相似度度量方法包括余弦相似度、调整余弦相似度、Pearson 相关系数和 Jaccard 相似度等[9]，以基于用户的协同过滤推荐算法为例。

2.2.1 余弦相似度

$$\mathrm{sim}_{\cos}(u,v)=\frac{\sum\limits_{i\in I_{u,v}} r_{u,i}r_{v,i}}{\sqrt{\sum\limits_{i\in I_{u,v}} r_{u,i}^2}\sqrt{\sum\limits_{i\in I_{u,v}} r_{v,i}^2}} \tag{2-1}$$

式中：$\mathrm{sim}_{\cos}(u,v)$——余弦相似度；

$I_{u,v}$——用户 u 和用户 v 的共同评分项目集；

$r_{u,i}$ 和 $r_{v,i}$——分别表示用户 u 和用户 v 对项目 i 的评分。

余弦相似度用两个向量的夹角余弦值度量相似度，两个向量的夹角越小，其夹角余弦值越大，则余弦相似度越高。这种方式对评分数值不敏感，如 $r_u=(1,1)$，$r_v=(5,5)$，$\text{sim}_{\cos}(u,v)=1$，尽管两个用户的评分差距很大（一个非常满意，一个非常不满意），但两个用户的相似度却为1。余弦相似度过于关注向量之间的夹角而忽视向量的长度（共同评分项数量），且过于依赖两个用户的共同评分。

2.2.2　调整余弦相似度

$$\text{sim}_{\text{acos}}(u,v)=\frac{\sum\limits_{i,j\in I_{u,v}}(r_{u,i}-\overline{r_u})(r_{v,j}-\overline{r_v})}{\sqrt{\sum\limits_{i\in I_u}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum\limits_{j\in I_v}(r_{v,j}-\overline{r_v})^2}} \tag{2-2}$$

式中：$\text{sim}_{\text{acos}}(u,v)$——调整余弦相似度；

$\overline{r_u}$——用户 u 的平均评分；

$\overline{r_v}$——用户 v 的平均评分。

调整余弦相似度通过减去平均值来提升对评分数值的敏感程度，但无法辨认其正负相关性。如 $r_u=(4,5)$，$r_v=(5,4)$，$\text{sim}_{\text{acos}}(u,v)=-1$，两个用户对项目的评价都非常满意，逻辑上是非常相似的，其计算结果与实际逻辑不相符。

2.2.3　Pearson 相关系数

$$\text{sim}_{\text{Pear}}(u,v)=\frac{\sum\limits_{i,j\in I_{u,v}}(r_{u,i}-\overline{r_u})(r_{v,j}-\overline{r_v})}{\sqrt{\sum\limits_{i,j\in I_{u,v}}(r_{u,i}-\overline{r_u})^2}\sqrt{\sum\limits_{i,j\in I_{u,v}}(r_{v,j}-\overline{r_v})^2}} \tag{2-3}$$

式中：$\text{sim}_{\text{Pear}}(u,v)$——Pearson 相关系数。

Pearson 相关系数是衡量两个数据集合之间的线性关系，令 $r_u=(1,1,3,3,2)$，$r_v=(3,5,5,5,4)$，$\text{sim}_{\text{Pear}}(u,v)=1$。两组数据的差异明显，用户 u 对项目的评分都较低，而用户 v 对项目的评分都较高，很难看出两者之间有相关性，其计算结果与理论分析完全相反。Pearson 相关系数考虑到用户评分的偏差，却忽略了用户共同评分的项目数，所以线性相关系数并不能完美地度量相似度。

2.2.4　Jaccard 相似度

$$\text{sim}_{\text{Jac}}(u,v)=\left|\frac{I_u\cap I_v}{I_u\cup I_v}\right| \tag{2-4}$$

式中：$\text{sim}_{\text{Jac}}(u,v)$——Jaccard 相似度；

I_u——用户 u 评分的项目；

I_v——用户 v 评分的项目。

Jaccard 相似度与 Pearson 相关系数不同，Jaccard 相似度仅考虑了两个用户的共同评分数，但未考虑绝对评分，从而影响用户相似度的准确性。

根据上述传统相似度计算方法存在的缺陷，通过改进相似度计算方法来缓解传统协同过滤推荐算法中存在的数据稀疏性问题，本书在基于近邻协同过滤推荐算法的基础上，提出了一种利用巴氏距离改进相似度度量方法，充分利用用户的每一个评分，即使没有一个共同项目评分时，也可以计算两个用户之间的相似度。Jaccard 相似度通过计算两个用户共同评分项目数的比重，能够在全局上衡量两个用户的相似度，以此作为全局相似度；Pearson 相关系数考虑到用户评分偏差，并结合巴氏系数得到的评分整体规律，以此可以作为局部相似度，将全局和局部相似度进行融合作为两个用户之间最终的相似度值。

2.3 一种巴氏系数改进相似度的协同过滤推荐算法

2.3.1 巴氏系数

巴氏系数(Bhattacharyya Coefficient，BC)是对两个统计样本的重叠的近似计算，可用来对两组样本的相关性进行测量，已广泛应用于信号处理、模式识别研究领域。在统计学中巴氏系数用于测量两种离散概率分布的可离性，衡量两个概率分布之间的相似度。

在连续域两个密度分布 $p_1(x)$和 $p_2(x)$之间的相似度用巴氏系数(BC)定义如式(2-5)所示。

$$\mathrm{BC}(p_1,p_2)=\int\sqrt{p_1(x)p_2(x)}\,\mathrm{d}x \tag{2-5}$$

在离散域上 BC 定义如式(2-6)所示。

$$\mathrm{BC}(p_1,p_2)=\sum_{x\in X}\sqrt{p_1(x)p_2(x)} \tag{2-6}$$

式中：密度 $p_1(x)$和 $p_2(x)$——已知的评分数据。

用户 u 和 v 的评分类别用 p_{uh} 和 p_{vh} 来评估其离散密度，计算用户 u 和 v 之间的 BC 相似度如式(2-7)所示。

$$\mathrm{BC}(u,v)=\mathrm{BC}(\hat{p}_u,\hat{p}_v)=\sum_{h=1}^{m}\sqrt{(\hat{p}_{uh})(\hat{p}_{vh})} \tag{2-7}$$

式中：m——评分类别总数；

$\hat{p}_{uh}=\dfrac{\#h}{\#u}$，其中$\#u$ 为对用户 u 评分的项目总量，$\#h$ 为用户 u 评分为 h 的项目总量，$\sum_{h=1}^{m}\hat{p}_{uh}=\sum_{h=1}^{m}\hat{p}_{vh}=1$。

例如，用户 u 和 v 的评分向量 $r_u=(1,0,2,0,1,0,2,0,3,0)^{\mathrm{T}}$，$r_v=(0,1,0,2,0,1,0,2,0,3)^{\mathrm{T}}$，评分在 1～5，所以用户 u 和 v 之间的 BC 系数如式(2-8)所示。

$$\begin{aligned}\mathrm{BC}(u,v) &= \sum_{h=1}^{5}\sqrt{\hat{p}_{uh}\,\hat{p}_{vh}} \\ &= \sqrt{\left(\frac{2}{5}\right)\times\left(\frac{2}{5}\right)}+\sqrt{\left(\frac{2}{5}\right)\times\left(\frac{2}{5}\right)}+ \\ &\quad \sqrt{\left(\frac{1}{5}\right)\times\left(\frac{1}{5}\right)}+0+0=1\end{aligned} \tag{2-8}$$

2.3.2　巴氏系数相似度

基于巴氏距离提出一种新的相似度度量方法——巴氏系数相似度，利用巴氏系数改进相似度根据用户的所有评分来计算两个用户之间的相似度，利用 Jaccard 相似度计算两用户之间的全局相似度，再利用巴氏系数分析评分分布并结合 Pearson 相关系数得到局部相似度，结合局部与全局相似度获得最终的相似度值。设用户 u 和 v 评价的两个项目分别为 I_u 和 I_v，当两用户之间的共同评分项目数为 0，即 $I_u \cap I_v = \varnothing$，用户 u 和 v 评价的两项目之间的相似度(用 BC 系数)和两个项目评分的局部相似度。其局部相似度定义如式(2-9)所示。

$$\mathrm{sim}_{\mathrm{loc}}(u,v) = \sum_{i\in I_u}\sum_{j\in I_v}\mathrm{BC}(u,v)\mathrm{loc}(r_{ui},r_{vj}) \tag{2-9}$$

式中：$\mathrm{BC}(u,v)$——用户 u 和 v 的全部评分信息；

$\mathrm{loc}(r_{ui},r_{vj})$——评分 r_{ui} 和 r_{vj} 之间的局部相似度。

BC 系数在没有共同评分情况下也可以计算用户 u 和 v 之间的相似度，如果两个用户在全局角度上是相似的，则 $\mathrm{BC}(u,v)$ 可以提高用户 u 和 v 的评分之间的局部相似度；另一方面，如果用户 u 和 v 完全不相似，则 $\mathrm{BC}(u,v)$ 降低两个用户评分之间局部相似度的重要性。局部相似度既提供非常重要的用户局部信息，也必须提供用户评分之间的正和负关系，如式(2-10)所示，可以用 $\mathrm{loc}(r_{ui},r_{vj})$ 表示 Pearson 相关系数来评估两评分之间的局部相似度。

$$\mathrm{loc}(r_{ui},r_{vj}) = \frac{\sum\limits_{i,j\in I_{u,v}}\left(r_{u,i}-\overline{r_u}\right)\left(r_{v,j}-\overline{r_v}\right)}{\sqrt{\sum\limits_{i,j\in I_{u,v}}\left(r_{u,i}-\overline{r_u}\right)^2}\sqrt{\sum\limits_{i,j\in I_{u,v}}\left(r_{v,j}-\overline{r_v}\right)^2}} \tag{2-10}$$

利用式(2-11)衡量两个用户评分之间的相关性 $\mathrm{sim}_{\mathrm{BC}}(u,v)$，融合局部相似度和全局相似度得到最终的相似度值。当 $\mathrm{BC}(u,v)=1$ 时局部相似度的作用最大，为相同项目提供最大的局部相似度；当 $\mathrm{BC}(u,v)=0$ 时局部相似度为 0，此时全局相似度就是最终的相似度值。

$$\mathrm{sim}_{\mathrm{BC}}(u,v) = \mathrm{sim}_{\mathrm{Jac}}(u,v) + \sum_{i\in I_u}\sum_{j\in I_v}\mathrm{BC}(u,v)\mathrm{loc}(r_{ui},r_{vj}) \tag{2-11}$$

本章提出的巴氏系数相似度特点在于整合利用巴氏系数、Jaccard 和 Pearson 相关系数互相弥补其缺陷，完美诠释两个用户之间的相似度；即使没有一个共同评分，也可以计算两个用户(项目)之间的相似度；计算相似度时利用用户对项目的所有评分，不会造成资源浪费；多方面考虑可能影响用户之间相似度的因素；考

虑用户评分的全局相似度和局部相似度。

2.3.3 BCCF 算法描述

在基于近邻协同过滤推荐算法的基础上，利用巴氏系数改进相似度计算，设计出利用巴氏系数改进相似度的协同过滤推荐算法 BCCF 流程。

算法 2-1 BCCF 算法

输入：用户——项目的评分矩阵 R，最近邻居数目 k。

输出：目标用户 u 对未知项目的预测评分集。

算法的基本流程如下：

步骤 1：由数据集中的训练集得到用户—项目评分矩阵 $R_{m\times n}$。

步骤 2：用式(2-4)计算用户 u 和用户 v 的全局相似度(当 $u=v$ 时令 $\text{sim}_{\text{Jac}}(u,v)=0$)。

步骤 3：利用式(2-7)得到用户 u 和用户 v 分别评价项目的评分分布规律(当 $u=v$ 时令 $\text{BC}(u,v)=0$)。

步骤 4：利用步骤 3 的计算结果，并通过式(2-9)计算用户 u 和用户 v 的局部相似度(当 $u=v$时令 $\text{sim}_{\text{loc}}(u,v)=0$)。

步骤 5：根据步骤 3 和 4 得到的全局相似度和局部相似度利用式(2-11)进行融合，形成最终的相似度。

步骤 6：利用步骤 5 计算任意两个用户之间的最终相似度，形成相似度矩阵，然后将目标用户 u 与其他用户 v 之间的最终相似度递减排序$\{\text{sim}(u,v_1)>\text{sim}(u,v_2)>\cdots>\text{sim}(u,v_m)\}$，选择相似度排序中前 k 个用户作为目标用户 u 的最近邻居，k 个最近邻居用户构成目标用户 u 的最近邻居集合 $I_u=\{v_1,v_2,\cdots,v_k\}$。

步骤 7：根据步骤 6 计算得到目标用户 u 的最近邻居集合 I_u，最近邻居用户对未知项目 i 的评分通过加权平均求和得到目标用户 u 对该项目的预测评分 $P_{u,i}$，如式(2-12)所示。

$$P_{u,i}=\overline{r_u}+\frac{\sum_{v=1}^{n}[\text{sim}(u,v)(r_{v,i}-\overline{r_v})]}{\sum_{v=1}^{n}\text{sim}(u,v)} \tag{2-12}$$

算法结束

2.4 实验与分析

2.4.1 数据集

为了检验提出方法的有效性，分别在 Movielens_100K 和 Movielens_1M 数据集上进行验证，数据集的具体参数如表 2-1 所列。

表 2-1 实验数据集的具体参数

名　称	用 户 数	项 目 数	评 分 总 数	稀 疏 度
Movielens_100K	943	1682	10^5	93.7%
Movielens_1M	6040	3706	10^8	95.81%

实验环境：Windows 7 操作系统，4GB 内存，Intel（R）Core（TM）i3-4150 CPU3.5GHz 开发工具使用 Matlab 2013b。

2.4.2　评价标准

本书采用 MAE 和 RMSE 作为度量标准。MAE 用来衡量推荐的精确率，能很好地反映预测值误差的实际情况。设在训练集上得到用户的预测评分集合为 $\{p_{u,1}, p_{u,2}, p_{u,3}, \cdots, p_{u,n}\}$，用户实际评分集合为 $\{r_{u,1}, r_{u,2}, r_{u,3}, \cdots, r_{u,n}\}$，则平均绝对误差 MAE 定义如式(2-13)所示。

$$\mathrm{MAE} = \frac{\sum_{u,i \in N} |r_{u,i} - p_{u,i}|}{N} \tag{2-13}$$

式中：$r_{u,i}$——用户 u 对项目 i 的真实评分；

$p_{u,i}$——用户 u 对项目 i 的预测评分；

N——测试集中用户评分的个数。

通过计算用户的预测评分和实际评分之间的偏差来度量算法预测的准确性，MAE 值越小，则预测评分与实际评分越接近，精确度越高。

均方根误差 RMSE 定义如式(2-14)所示。

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i \in N} |r_{u,i} - p_{u,i}|^2}{N}} \tag{2-14}$$

RMSE 同样用来衡量推荐的精确率，RMSE 更侧重于预测评分与实际评分差值的绝对值，相对 MAE 加大了惩罚力度，RMSE 值越小，则 MAE 值越小，推荐精确度越高。

2.4.3　实验结果与分析

本实验随机将 Movielens 数据集按 8∶2 比例分为训练集和测试集，并进行 10 次交叉实验求得平均值作为最终结果。

1．在 Movielens_100K 数据集上传统相似度比较

为了验证和评估本书提出的基于巴氏系数相似度对于邻居数目的变化情况，将巴式系数相似度与传统相似度的 MAE 进行比较，如图 2-1 所示。

从图 2-1 可以看出，当邻居数目 k 在 5～30 变化时，比较 Pearson 相关系数、Jaccard 相似度的 MAE 值。Jaccard 相似度在全局上计算共同评分所占比重，由巴氏系数取得评分分布并结合 Pearson 相关系数在局部上得到评分之间的相似度，由全部和局部相似度得到巴氏系数相似度。从图 2-1 中可以看出，随着用户邻居数目 k 的增大，本书提出的 BCCF 算法其 MAE 值总体趋于减少，当邻居数达到 30 时 MAE 值基本稳定；相对于传统的 Jaccard 和 Pearson 相关系数计算方法，BCCF 算法的 MAE 值最低且相对稳定。

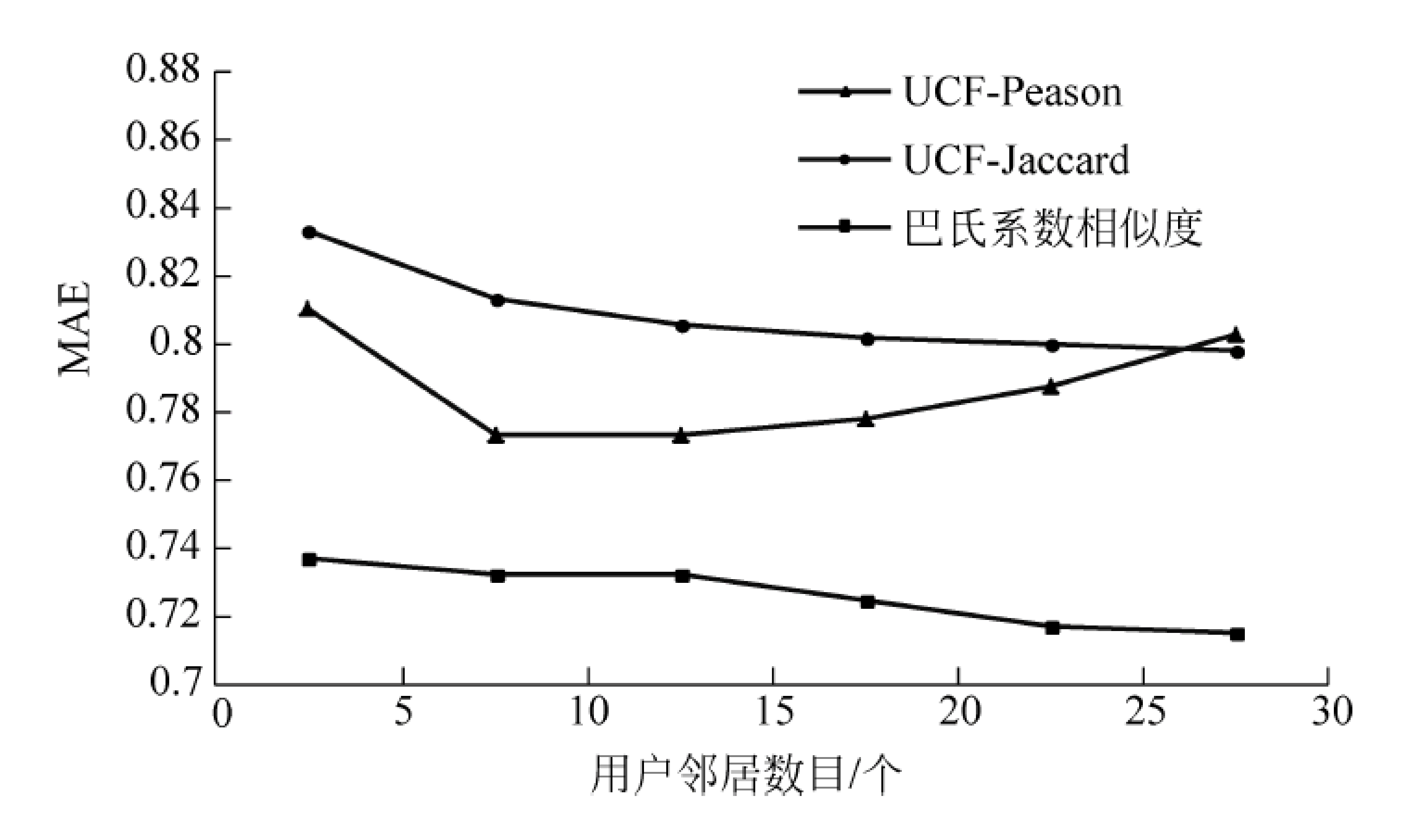


图 2-1 巴式系数相似度与传统相似度的 MAE 比较

2. 在 Movielens_100K 数据集上不同算法之间对比

为了验证同一数据集上不同算法之间 MAE 和 RMSE 的变化对比情况，如图 2-2 和图 2-3 所示分别为 Movielens_100K 数据集上 3 种算法的 MAE 对比和 Movielens_100K 数据集上 3 种算法的 RMSE 对比，实验中最近邻居数 k 分别取 5、10、15、20、25 和 30，将巴氏系数相似度作为算法的相似度计算方法，本章提出的 BCCF 算法分别与文献[7]中提出的改进相似度算法（JPCC）、文献[8]中提出的改进算法（JMSD）在 Movielens_100K 数据集上进行 MAE 和 RMSE 值对比。

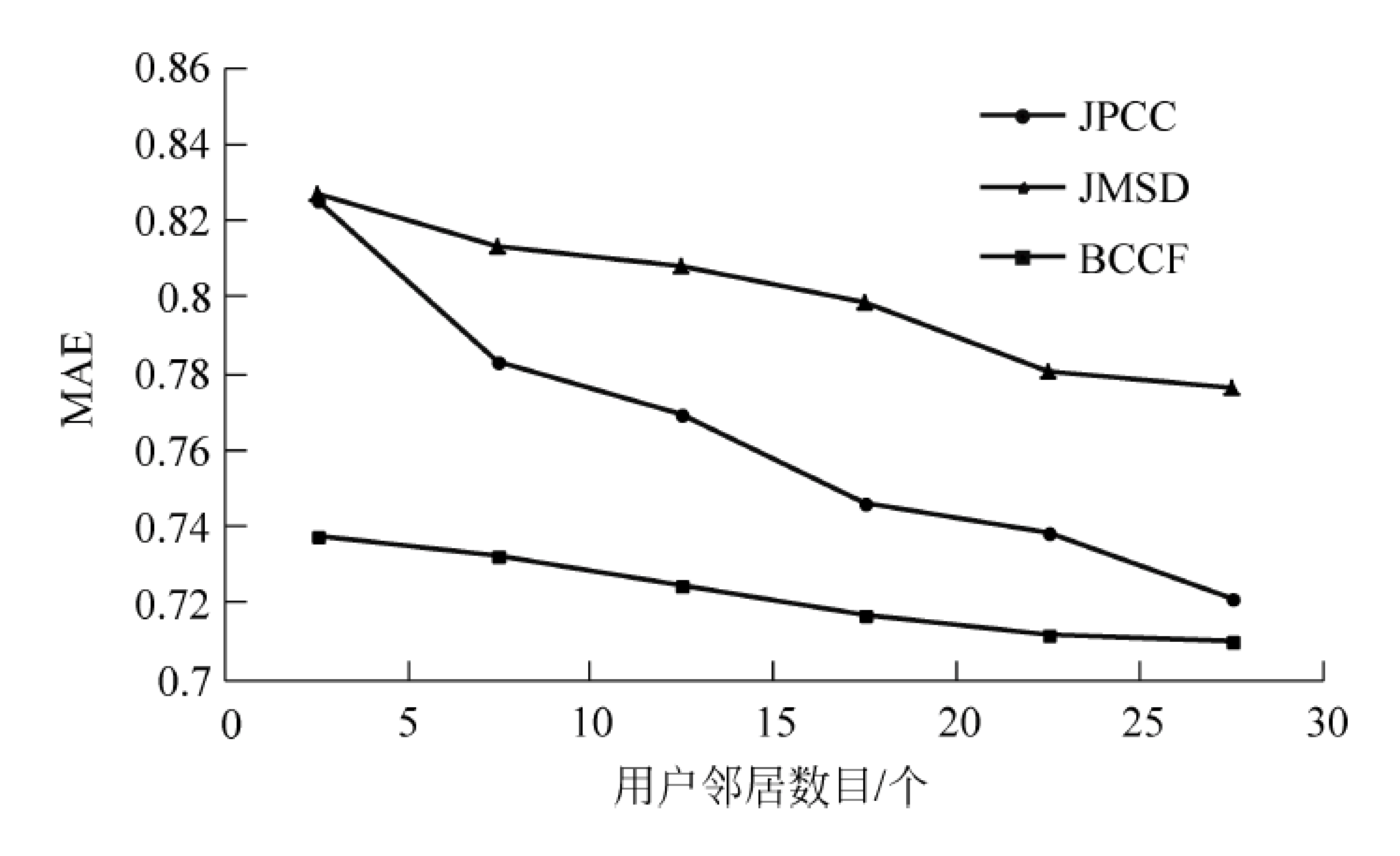


图 2-2 Movielens_100K 数据集上 3 种算法的 MAE 对比

由图 2-2 和图 2-3 可以看出，在 Movielens_100K 数据集上改进算法 BCCF 与传统的 JPCC 算法相比 MAE 在趋于稳定状态下降低了 6%，RMSE 降低了 2%；比 JMSD 算法相比 MAE 在趋于稳定状态下降低了 10%，RMSE 降低了 5%。在 Movielens_100K 数据集上 BCCF 算法的精确性比上述所参照的经典算法的精确度有所提高，从一个侧面说明了在该数据集中实际存在着用户对项目的共同评分很少的情况下进行优化的方案，即采用巴氏系数进行用户间所有的评分能够进一步提高算法的精度。

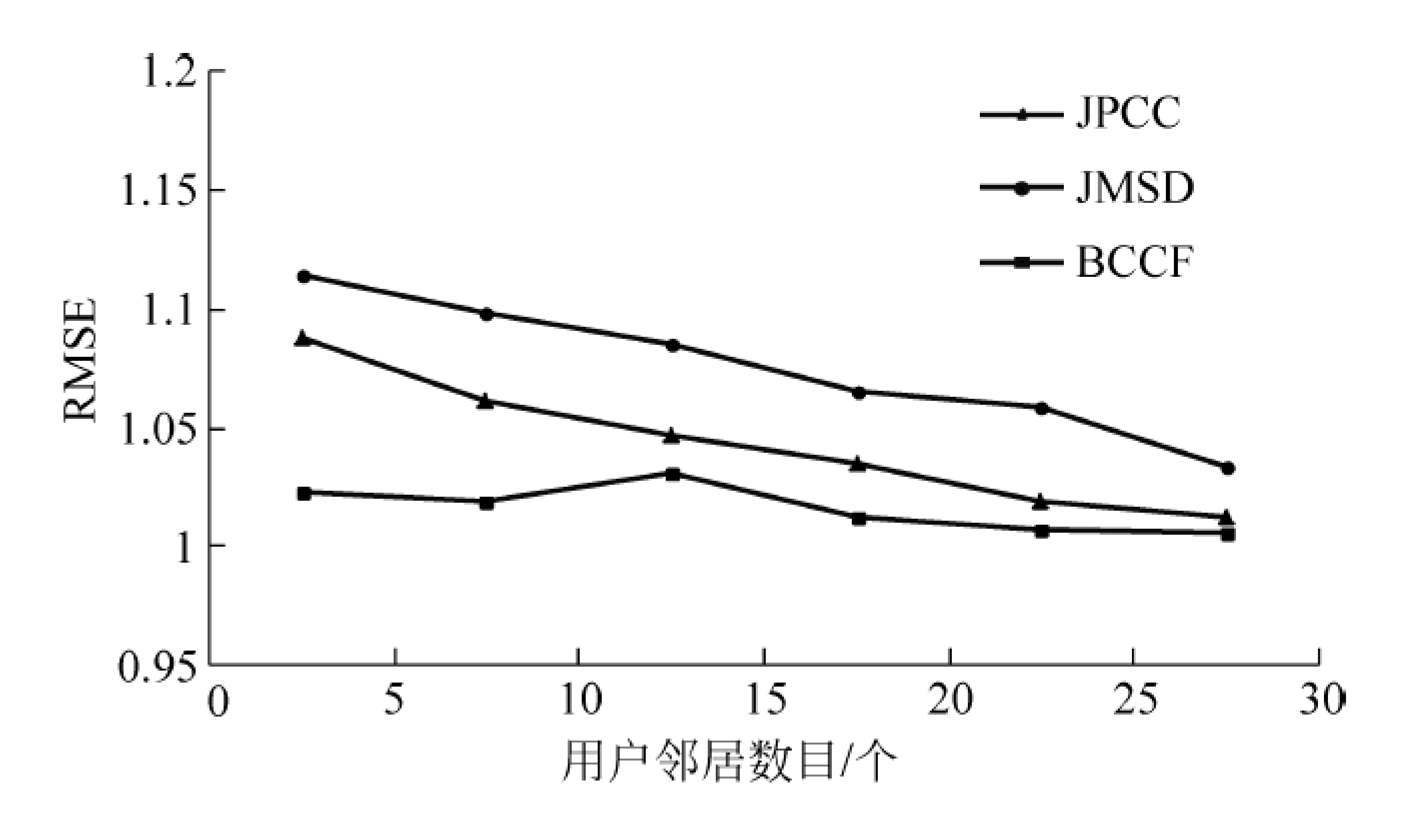


图 2-3　Movielens_100K 数据集上 3 种算法的 RMSE 对比

3. 在 Movielens_1M 数据集上不同算法之间对比

本节实验中最近邻居数 k 为 5～30，将巴氏系数改进相似度作为算法的相似度计算方法。如图 2-4 所示为 Movielens_1M 数据集上 3 种算法的 MAE 对比，图 2-5 所示为 Movielens_1M 数据集上 3 种算法的 RMSE 对比，并将本章提出的 BCCF 算法分别与 JPCC 算法、JMSD 算法进行了 MAE 和 RMSE 值对比。

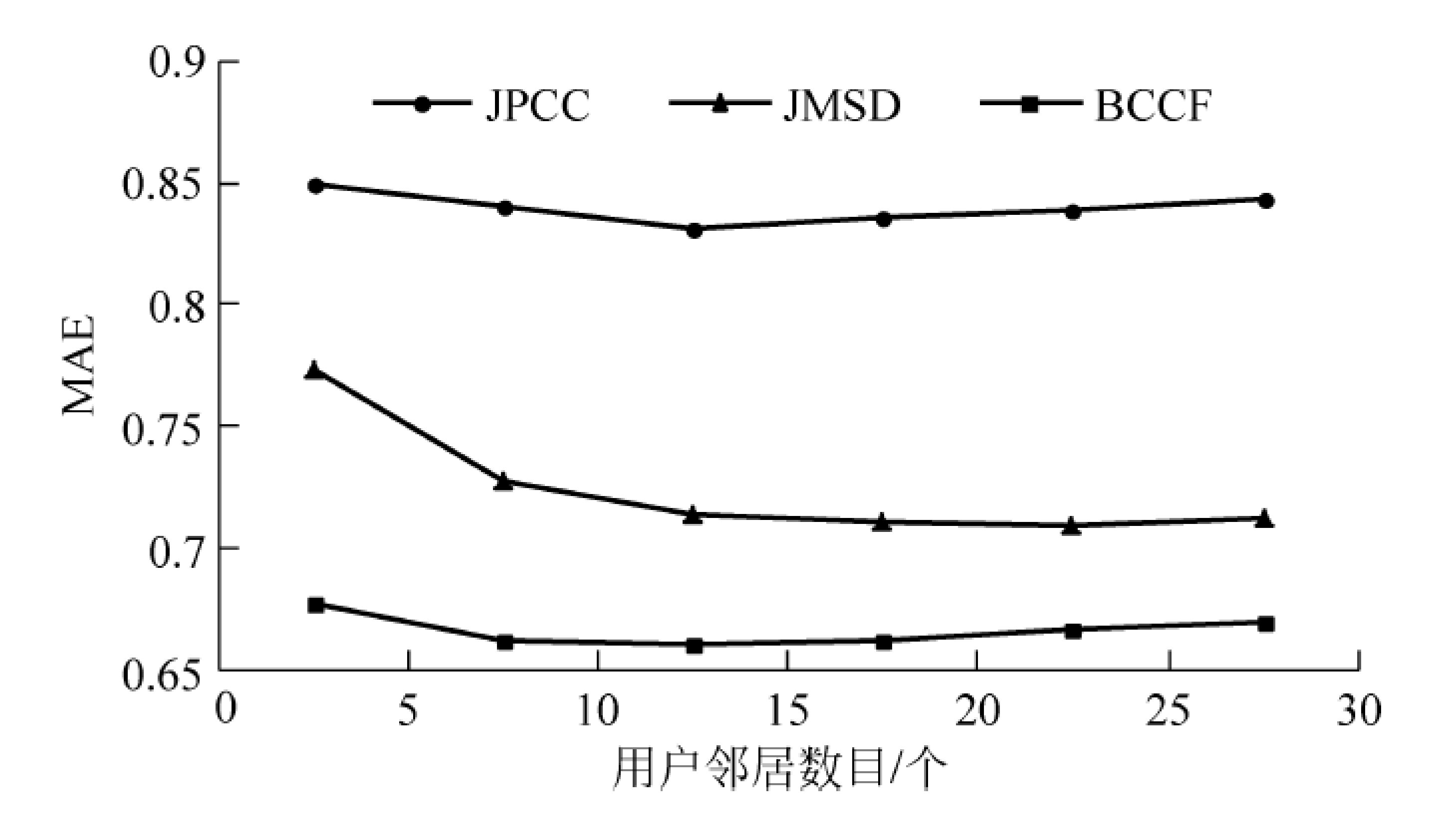


图 2-4　Movielens_1M 数据集上 3 种算法的 MAE 对比

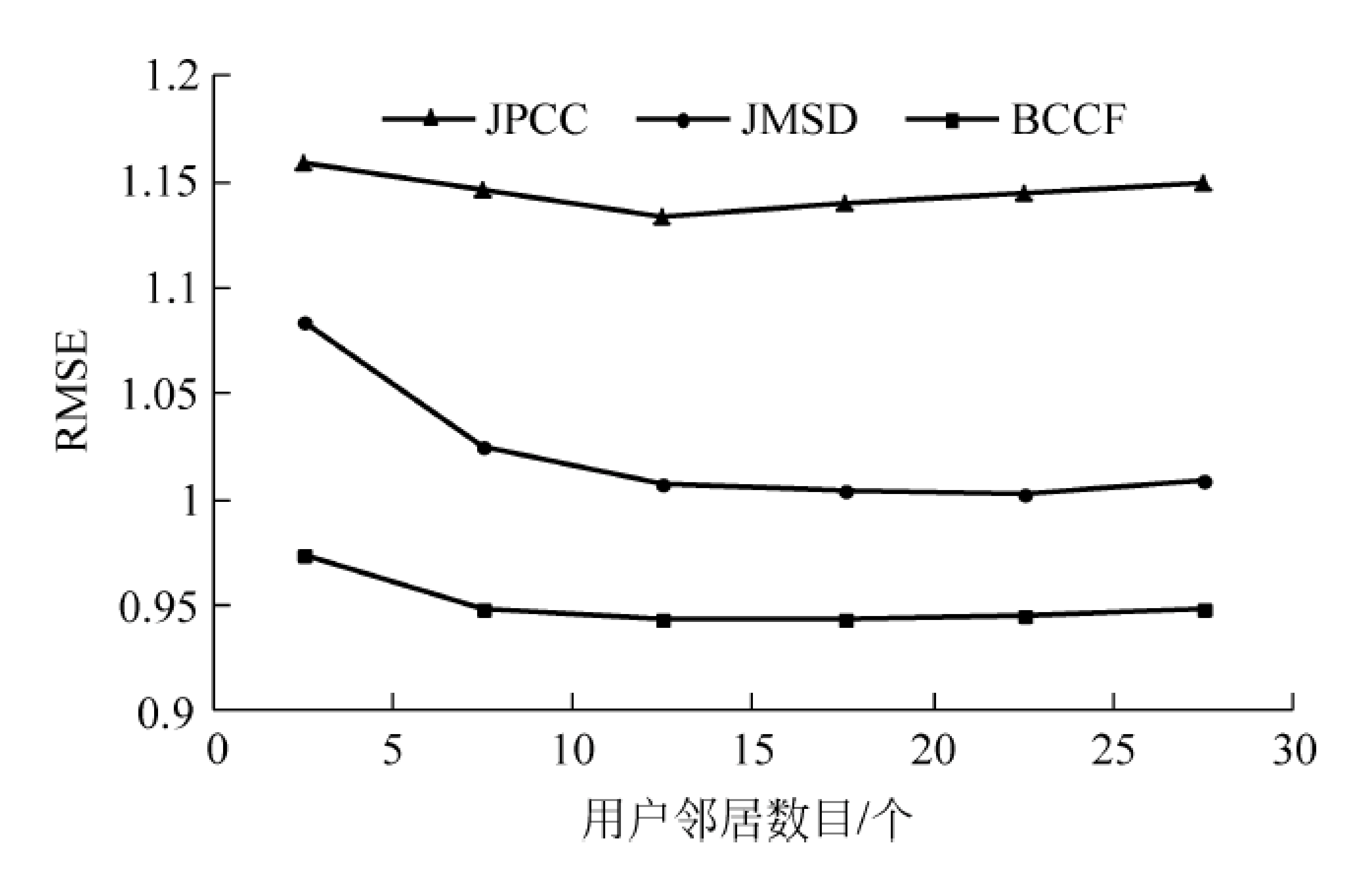


图 2-5　Movielens_1M 数据集上 3 种算法的 RMSE 对比

由图 2-4 和图 2-5 可以看出，在 Movielens_1M 数据集上改进算法 BCCF 与传统的 JPCC 算法相比在趋于稳定状态下 MAE 降低了 20%，RMSE 降低了 17%，比 JMSD 算法相比在趋于稳定状态下 MAE 降低了 10%，RMSE 降低了 4%。由于 RMSE 和 MAE 值越小则推荐精度越高，且在 Movielens_1M 数据集上 BCCF 算法的 RMSE 和 MAE 值比在 Movielens_100K 数据集上实验结果更小，说明数据集越稀疏其 BCCF 算法的 MAE 和 RMSE 性能越好，说明本书提出的巴氏系数改进相似度的协同过滤推荐算法能有效缓解评分数据稀疏问题及其带来的推荐精度问题。

本章小结

本章提出了一种基于巴氏系数的改进相似度计算方法，利用用户的每一个评分，考虑了用户评分的全局和局部相似度，能更全面分析用户之间的相似度，并在此基础上设计了 BCCF 算法，通过与传统协同过滤推荐算法比较，实验结果表明：BCCF 不仅仅依赖于两个用户的共同评分，而是利用两个用户的所有评分，现实的系统中用户对项目的共同评分很少甚至没有，故 BCCF 有很强的实用性；BCCF 合理利用用户的每一个评分，根据用户的评分来发现其内在的相关性；BCCF 更适用于评分矩阵稀疏的数据集。考虑用户的所有评分，造成计算相似度时其时间复杂度增加，对巴氏系数相似度计算进行优化，并优化巴氏系数相似度中局部相似度信息量计算部分，找到能降低其时间复杂度的方法，进一步提高算法性能。

参考文献

[1] 林耀进，张佳，林梦雷，等. 一种基于模糊信息熵的协同过滤推荐方法[J]. 山东大学学报(工学版)，2016(05)：13-20.

[2] 王俊，李石君，杨莎. 一种新的用于跨领域推荐的迁移学习模型[J]. 计算机学报，2017(10)：2367-2380.

[3] Do Thi Lien，Nguyen Duy Phuong. Collaborative Filtering with a Graph-based Similarity Measure[C]. Computing，Management and Telecommunications (ComManTel)，2014 International Conference on. IEEE. Da Nang，2014，251-256.

[4] Bobadilla J，Ortega F，Hernando A，et al. Recommender Systems Survey[J]. Knowledge-Based Systems，2013，46 (1)：109-132.

[5] Xiao Han，Leye Wang，Reza Farahbakhsh，et al. CSD：A Multi-User Similarity Metric for Community Recommendation in Online Social Networks [J]. Expert Systems with Applications，2016，53：14-26.

[6] Rong Hui-gui，Huo Sheng-xu，Hu Chun-hua，et al. User Similarity-based Collaborative Filtering Recommendation Algorithm [J]. Journal on Communications，2014，35(2)：16-24.

[7] Wu Yi-tao，Zhang Xing-ming，Wang Xing-mao，et al. User Fuzzy Similarity-based

Collaborative Filtering Recommendation Algorithm [J]. Journal on Communications, 2014, 35(2): 16-24.

[8] Sun Da-ming, Zhang Bin, Zhang Shu-bo, et al. A Popularity Versus Similarity Query Recommendation Strategy [J]. Journal of Chinese Computer Systems, 2016, 37 (6): 1121-1125.

[9] Zheng Cui-cui, Lin Li. Research on Method of Similarity Measurement in Collaborative Filter Algorithm [J]. Computer Engineering and Applications, 2014, 50(8): 147-149.

[10] Jesus Bobadilla, Fernando Ortega, Antonio Hernando, et al. A Similarity Metric Designed to Speed Up, Using Hardware, the Recommender Systems *k*-nearest Neighbors Algorithm [J]. Knowledge-Based Systems, 2013, 51(1): 27-34.

[11] Parivash Pirasteh, Dosam Hwang, Jai E Jung. Weighted Similarity Schemes for High Scalability in User-based Collaborative Filtering [J]. Mobile Networks & Applications, 2015, 20(4): 497-507.

[12] Sun Hui-feng, Chen Jun-liang, Yu Guang, et al. JacUOD: A New Similarity Measurement for Collaborative Filtering[J]. Journal of Computer Science & Technology, 2012, 27(6): 1252-1260.

[13] Thorat P B, Goudar R M, Barve S. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System [J]. International Journal of Computer Applications, 2015, 110(4): 31-36.

[14] Lops P, de Gemmis M, Semeraro G, et al. Content-based and Collaborative Techniques for Tag Recommendation: an Empirical Evaluation [J]. Journal of Intelligent Information Systems, 2013, 40(1): 41-61.

[15] Goldberg D, Oki B M, Oki B M, et al. Using Collaborative Filtering to Weave an Information Tapestry[J]. Communications of the Acm. , 1992, 35(12): 61-70.

[16] Wolff J G. A Scaleable Technique For Best-Match Retrieval of Sequential Information Using Metrics-Guided Search[J]. Journal of Information Science, 1994, 20(1): 16-28.

[17] Pham X H, Nguyen T T, Jung J J, et al. -Spear: A New Method for Expert Based Recommendation Systems[J]. Journal of Cybernetics, 2014, 45(2): 165-179.

[18] Guo G, Zhang J, Thalmann D. Merging Trust in Collaborative Filtering to Alleviate Data Sparsity and Cold Start[J]. Knowledge-Based Systems. 2014, 57(2): 57-68.

[19] Ortega F, Hernando A, Bobadilla J, et al. Recommending Items to Group of Users Using Matrix Factorization based Collaborative Filtering[J]. Information Sciences, 2016, 345(C): 313-324.

[20] Zhou X, He J, Huang G, et al. SVD-based Incremental Approaches for Recommender Systems[J]. Journal of Computer & System Sciences, 2015, 81(4): 717-733.

[21] Zhou T, Shan H, Banerjee A, et al. Kernelized Probabilistic Matrix Factorization: Exploiting Graphs and Side Information[J]. SDM, 2012.

[22] Vozalis M G, Margaritis K G. Applying SVD on Item-based Filtering [C]//null. IEEE Computer Society, 2005: 464-469.

[23] 刘庆鹏，陈明锐. 优化稀疏数据集提高协同过滤推荐系统质量的方法[J]. 计算机应用，2012, 4, (4): 1082-1085.

[24] Chen W. Multi-Collaborative Filtering Trust Network For Online Recommendation[J].

Information Systems Frontiers,2015,15(4):533-551.

[25] Chen C,Zeng J,Zheng X,et al. Recommender System-based on Social Trust Relationships [C]//e-Business Engineering (ICEBE),2013 IEEE 10th International Conference on. IEEE,2013:32-37.

[26] Moradi P,Ahmadian S,Akhlaghian F. An Effective Trust-based Recommendation Method Using a Novel Graph Clustering Algorithm[J]. Physica A Statistical Mechanics & Its Applications,2015,436:462-481.

[27] 原福永,蔡红蕾,李莉. 加入用户偏好的非均匀资源分配推荐算法[J]. 小型微型计算机系统,2015,36(2):205-210.

[28] 燕彩蓉,张青龙,赵雪,等. 基于广义高斯分布的贝叶斯概率矩阵分解方法[J]. 计算机研究与发展,2016,52(12):2793-2800.

[29] 王东. 基于贝叶斯网络的在线社交网络推荐技术研究[D]. 南京:南京邮电大学,2016.

[30] 赵海燕,熊波,陈庆奎,等. 基于信任传播的概率矩阵分解算法[J]. 小型微型计算机系统,2016,37(5):895-901.

[31] 郭弘毅,刘功申,苏波,等. 融合社区结构和兴趣聚类的协同过滤推荐算法[J]. 计算机研究与发展,2016,53(8):1664-1672.

[32] Pirasteh P,Hwang D,Jung J J. Exploiting Matrix Factorization to Asymmetric User Similarities in Recommendation Systems[J]. Knowledge-Based Systems,2015,83(1):51-57.

[33] Koren Y. Collaborative Filtering with Temporal Dynamics[J]. Communications of the ACM,2010,53(4):89-97.

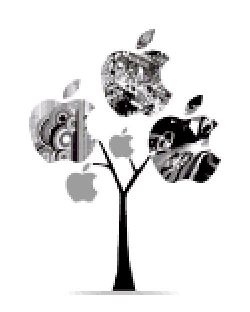
[34] Levy O,Goldberg Y. Neural Word Embedding as Implicit Matrix Factorization[J]. Advances in Neural Information Processing Systems,2014,3:2177-2185.

[35] Grasedyck L,Kluge M,Krämer S. Variants of Alternating Least Squares Tensor Completion in the Tensor Train Format[J]. Siam Journal on Scientific Computing,2015,37(5):A2424-A2450.

[36] Finlayson G D,Darrodi M M,Mackiewicz M. The Alternating Least Squares Technique for Nonuniform Intensity Color Correction[J]. Color Research & Application,2015,40(3):232-242.

第 3 章

基于用户兴趣和项目属性的协同过滤推荐算法



针对传统协同过滤推荐算法不能及时反映用户的兴趣变化、时效性不足导致推荐精度不高的问题，提出一种基于用户兴趣和项目属性的协同过滤推荐算法。在传统协同过滤基础上综合考虑评分时间、相似度以及项目属性等因素，首先在计算相似度过程中加入基于时间的用户兴趣度权重函数，然后再与项目属性相似度进行融合，最后进行项目预测与推荐。在 Movielens 数据集上的实验结果表明，所提出的算法与已有的经典算法相比，平均绝对误差降低了 3%～6%，有效提高了推荐的准确性。

3.1 引言

个性化推荐是从海量数据中挖掘出有用信息的一种技术，协同过滤是其应用最广泛、最成功的推荐算法之一，通过收集和分析用户的信息数据来学习用户的兴趣偏好和行为模式，从而为用户推荐所需要的信息或商品。

传统的协同过滤推荐算法忽略了随着时间变化而用户的兴趣也在不断发生变化的问题，即存在用户兴趣漂移现象。用户的兴趣偏好不但范围广泛，而且实时变化。例如，一个孩子在几岁时可能对动画片感兴趣，青春期可能对浪漫爱情片感兴趣，随后有可能对文艺片感兴趣，再过几年可能对剧情片感兴趣，等等。随着时间推移，用户的关注点在不断变化，如何捕获这一动态的时间效应是个难题。

通常将时间窗作为判断用户兴趣变化的一种表征方式，采用加权处理的方法，来提高推荐质量。文献[1]通过对心理遗忘曲线拟合出用户兴趣权重函数，提出基于时间窗的改进协同过滤推荐算法，从而追踪和学习用户的兴趣偏好；文献[2]～[4]提出基于评价时间数据权重的用户兴趣度量函数，使得用户最可能感兴趣近期访问过的资源。这些方法在相似度度量过程中加入了时间因子，从一定程度上解决了用户兴趣漂移问题，但是忽略了不同对象的类别属性等特征信息，这在一定程度上也会影响最终的推荐质量。

针对这一问题，本章提出了一种基于用户兴趣和项目属性的协同过滤推荐算

法，在传统的用户—项目评分矩阵基础上综合考虑用户偏好、评分时间以及项目属性特征等因素，先在计算相似度过程中加入时间逻辑性因素，再与项目属性相似度进行融合，明确用户对项目中各个属性的偏好程度，更能体现出用户的行为需求。

3.2 相关工作

提供个性化推荐服务已经成为各大电子商务网站和社交媒体的核心竞争力所在，如何实时根据用户的浏览和购买行为为其推荐更加符合用户偏好的项目，即在正确的时间推荐合适的项目，是目前推荐系统面临的一大挑战。在信息飞速发展的当代，用户兴趣和信息话题的流行转移速度也非常快。文献[5,6]提出了一个实时的在线推荐系统——TencentRec，并在此系统上部署一系列的应用，每天为10 亿用户根据其兴趣爱好实时推荐话题，在实践中观察 TencentRec 的性能；并在此系统上提出了一个基于项目的可扩展协同过滤推荐算法，处理隐式反馈问题，通过增量更新和实时修剪以减少计算成本，对数据进行实时采集和处理，可以随时捕捉用户的兴趣，提高推荐质量。文献[7]提出了一种基于空间正则化和突发加权平滑的混合模型推荐算法，利用正则化框架发现社交网络中的空间信息以及在时间轴上采用突发加权平滑方案发现的时变信息，实验结果表明所提出的混合模型能够在单一的检测过程中区别时变话题和稳定话题，从而可以针对用户不同的兴趣分别推荐话题。此算法只适用于新闻类信息网站，针对目前流行的社交媒体平台，文献[8]设计了一个潜在的类统计混合模型，称为时间上下文感知混合模型(TCAM)，TCAM 同时根据用户内在兴趣相关话题和时间上下文相关话题这两个因素的影响对用户行为建模；为了进一步提高 TCAM 算法的性能，提出项目加权方法使 TCAM 更好地为用户推荐其偏好的项目。文献[9]提出了一种利用人类行为对信息过滤的协同过滤推荐算法，与传统的协同过滤推荐算法相比推荐精度得到了很大提高，同时改善了推荐的新颖性和多样性。上述研究在计算相似度时加入时间因素，但是没有考虑项目特征属性问题，为此本书提出一种基于用户兴趣和项目属性的协同过滤推荐算法，在计算相似度时不单单考虑用户兴趣而且考虑项目本身的属性特征，提高算法的推荐精度。

时间是一种重要的上下文信息，对用户兴趣偏好有着深入的影响。本书以 Movielens 数据集为例(1997 年 9 月—1998 年 4 月，以月为单位)，分析了 3 种不同类型影片观众人次比例随时间的变化情况，如图 3-1 所示为 3 种不同类型影片月观众人次比例走势图。

图 3-1 中 Item1 属于喜剧动画片，Item50 属于战争科幻动作片，Item181 属于浪漫喜剧片。从图 3-1 可以看出，不同类型影片，随着时间的变化，影片受欢迎的程度发生改变，观影人次也相应发生变化。不管是属于哪种类别的电影，其观众人次比例随时间的推移都在逐渐下降，其规律大体上与心理学上的遗忘曲线相似：

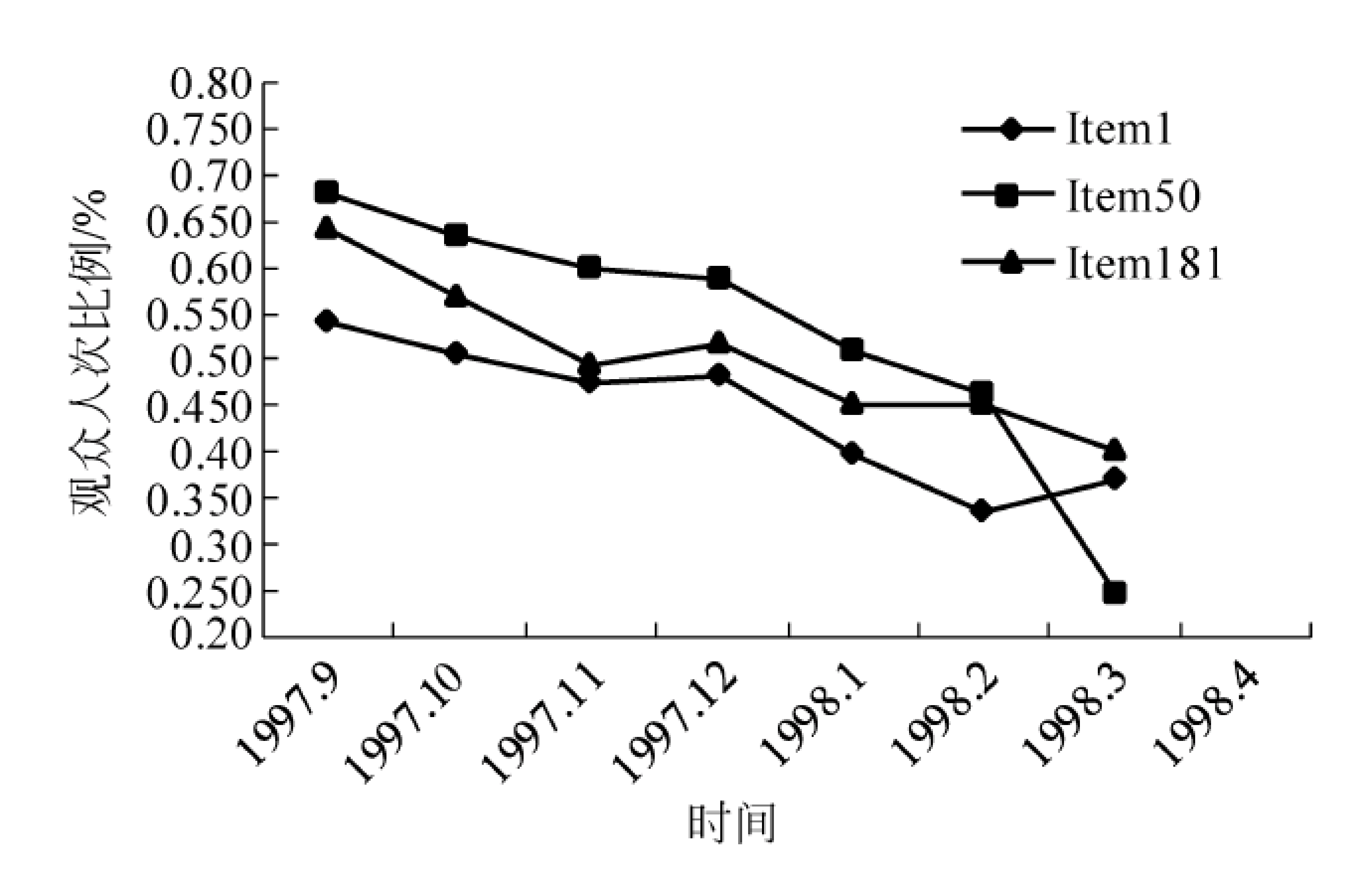


图 3-1 3 种不同类型影片月观众人次比例走势图

电影刚上映时关注的观众比较多，随着时间的推移观众人次慢慢下降，直到被人们所遗忘。这也符合 Ebbinghaus 遗忘曲线的规律，即人类记忆能力随时间的变化而降低。

传统的协同过滤推荐算法分为输入数据、寻找最近邻居集合和预测推荐 3 步。寻找最近邻居集合是协同过滤推荐算法中关键的一步，其结果直接影响推荐的准确度。寻找最近邻居集合可通过计算相似度方法实现，常用的相似度度量方法有余弦相似度、调整余弦相似度和 Pearson 相关相似度。

3.3 基于用户兴趣和项目属性的协同过滤推荐算法

3.3.1 基于时间的用户兴趣度权重

本章采用拟合的遗忘曲线对项目评分进行时间加权，离采样时间越近的评分赋予较大的权值，反之则赋予较小的权值，以此来模拟用户的兴趣爱好随着时间而不断变化。因此，可以根据遗忘曲线定义指数衰减函数表示用户兴趣的变化，基于时间的用户兴趣度权重函数如式(3-1)所示。

$$w_t = e^{-\frac{t_{ui}-t_0}{T}} \tag{3-1}$$

式中：t_{ui}——用户 u 对项目 i 的评分时间；

t_0——目标用户的采样时间；

T——整个数据集的时间跨度(结束时间—开始时间)。

本章以 Pearson 相关相似度作为相似计算公式，并将基于时间的用户兴趣度权重函数引入相似度计算公式中，基于用户兴趣度权重的 Pearson 相关相似度计算方法如式(3-2)所示。

$$\mathrm{sim}(i,j) = \frac{\sum_{c\in I_{ij}} w_t (r_{c,i} - \overline{r_i})(r_{c,j} - \overline{r_j})}{\sqrt{\sum_{c\in I_{ij}} (r_{c,i} - \overline{r_i})^2} \sqrt{\sum_{c\in I_{ij}} (r_{c,j} - \overline{r_j})^2}} \tag{3-2}$$

式中：$r_{c,i}$——用户 c 对项目 i 的评分；

$r_{c,j}$——用户 c 对项目 j 的评分；

$\overline{r_i}$——项目 i 的平均评分；

$\overline{r_j}$——项目 j 的平均评分。

3.3.2 改进相似度计算

加入兴趣度权重能够有效地找出用户的喜好范围。为了更好地进行推荐服务，需要找出用户偏好的特征属性，避免把一个用户不喜欢的影片当成是用户喜好的影片进行推荐。在考虑时间效应的基础上计算出用户评分与项目属性之间的关系，发现用户对于项目中各个属性的喜好程度，结合基于时间的用户兴趣度权重和项目属性能够明确用户的兴趣偏好，准确有效地找出其邻居集合。

令项目属性的集合为 $l=\{l_1, l_2, \cdots, l_d\}$，其中 d 为类别属性个数。以 Movielens 数据集为例：数据集中的电影一共有 18 个类别属性，分别为{unknown，Action，Adventure，…，Western}。项目属性特征可以用一个 $n\times d$ 的项目属性矩阵 $G_{n\times d}$ 来计算，其中 n 为项目个数，g_{id} 为 0 时代表项目 i 不具有这个属性，g_{id} 为 1 时代表项目 i 具有该属性。项目的特征属性相似度计算方法如式(3-3)所示。

$$\mathrm{Esim}(i,j)=\frac{(i_l\cdot j_l)}{(||i_l|\cdot|i_l||+||j_l|\cdot|j_l||-i_l\cdot j_l)} \tag{3-3}$$

式中：i_l——项目 i 的属性集合；

j_l——项目 j 的属性集合。

项目之间相似度不能单一通过用户对项目的评分来计算，还要考虑项目与项目之间的相关相似度，采用算术加权平均，综合考虑项目的特征属性相似度和评分相似度，获得更全面的相似度度量模型，融合的相似度计算方法如式(3-4)所示。

$$\mathrm{sim}(i,j)=\gamma\times\mathrm{Esim}(i,j)+(1-\gamma)\times\mathrm{sim}(i,j) \tag{3-4}$$

式中：γ——平衡因子用作协调两方面相似度度量的结果，$0<\gamma<1$。

γ 在[0,1]中取一系列值，观察不同 γ 值对推荐准确度的影响，选择合适的 γ 值将两种相似度进行融合，提高推荐准确率。

3.3.3 加权预测评分

由相似度计算得到最近邻居集合后，考虑时间对预测值的影响，用户现在的行为应该和用户最近的行为关系更大。将基于时间的用户兴趣度权重 w_t 加入到预测评分中，其计算方法如式(3-5)所示。

$$P_{c,i}=\overline{r_i}+\frac{\sum_{j=1}^{n}\left[\mathrm{sim}(i,j)\left(r_{c,j}\times w_t-\overline{r_j}\right)\right]}{\sum_{j=1}^{n}\mathrm{sim}(i,j)} \tag{3-5}$$

式中：$\mathrm{sim}(i,j)$——目标项目 i 与最近邻居项目 j 的相似度度量；

$r_{c,j}$——用户 c 对项目 j 的评分；

$\overline{r_i}$——项目 i 的平均评分；

$\overline{r_j}$——项目 j 的平均评分。

为了有效计算出用户的当前兴趣，改进的预测评分用 w_t 赋予评分矩阵中每个评分一个权重，即用户最近数据贡献度更大，占较大的权重，反之亦然。

3.3.4　算法步骤

在基于近邻协同过滤推荐算法的基础上，在计算相似度过程中加入基于时间的用户兴趣度权重函数，然后再与项目属性相似度进行融合，最后进行项目预测与推荐。基于用户兴趣和项目属性的协同过滤推荐算法流程如算法 3-1 所示。

算法 3-1　基于用户兴趣和项目属性的协同过滤推荐算法

输入：数据集中的一对训练集和测试集，最近邻居个数 neighbor_num，平衡参数 γ。

输出：用户 c 对测试集中项目 i 的预测评分。

算法的基本流程如下：

步骤 1：由训练集得到用户—项目评分矩阵 $R_{m\times n}$ 和时间矩阵 $T_{m\times n}$。例如：用户 1 对项目 1 的评分为 5 且评分时间为 874 965 758，则在评分矩阵 $R_{m\times n}$ 中 $r_{1,1}=5$，时间矩阵 $T_{m\times n}$ 中 $t_{1,1}=874\ 965\ 758$。

步骤 2：利用用户兴趣度权重函数 w_t，计算目标用户的兴趣度权重。

步骤 3：用式(3-5)计算项目 i 和项目 j 的评分相似度(当 $j=i$ 时令 $\mathrm{sim}(i,j)=0$)。

步骤 4：通过式(3-6)计算项目 i 和项目 j 的特征属性相似度(当 $j=i$ 时令 $\mathrm{sim}(i,j)=0$)。

步骤 5：利用式(3-7)进行融合对根据步骤 3 和 4 得到的评分相似度和项目属性相似度，形成最终的相似度矩阵。

步骤 6：根据步骤 5 计算得到的相似度矩阵来寻找目标项目 i 的最近邻居，邻居关系的计算是为了对每一个项目 i 找到一个邻居集合 $\mathrm{Neighbor}_i=\{j_1,j_2,\cdots,j_m\}$，$i\notin \mathrm{Neighbor}$，将相似度$\{\mathrm{sim}(i,j_1)>\mathrm{sim}(i,j_2)>\cdots>\mathrm{sim}(i,j_m)\}$递减排序。根据预先设定的邻居数 neighbor_num，选择 $\mathrm{sim}(i,j)$ 最大的前 neighbor_num 个作为项目 i 最近邻居。

步骤 7：根据步骤 6 得到的目标项目 i 的最近邻居集合 $\mathrm{Neighbor}_i$ 和评分矩阵 $R_{m\times n}$ 中的评分，依据式(3-8)利用用户 c 对项目 i 的最近邻居评分来预测用户 c 对项目 i 的评分。

算法结束

3.4　实验结果与分析

3.4.1　数据集

本章选用 MovieLens 数据集对提出的算法在 Matlab 中进行评估测试，该数据集包含 943 个用户对 1682 部电影连续 7 个月左右的评分数据，评分范围是 1～5，1 表示“很差”，5 表示“很好”。整个数据集的稀疏等级为 1－100 000/(943×1682)＝93.7%。

MovieLens 数据集提供 5 组随机划分的训练集和测试集，实验在这 5 组数据上分别进行，最终实验结果为这 5 次结果的算术平均值。

3.4.2 评价标准

本章采用平均绝对误差（Mean Absolute Error，MAE）衡量推荐的精确率，能更好地反映预测值误差的实际情况。设在训练集上得到用户的预测评分集合为 $\{p_{u,1}, p_{u,2}, p_{u,3}, \cdots, p_{u,n}\}$，用户实际评分集合为 $\{r_{u,1}, r_{u,2}, r_{u,3}, \cdots, r_{u,n}\}$，则平均绝对误差 MAE 定义如式（3-6）所示。

$$\text{MAE} = \frac{\sum_{u,i \in N} \mid r_{u,i} - p_{u,i} \mid}{N} \tag{3-6}$$

式中：N——测试集中用户评分的个数；

$p_{u,i}$——用户 u 对第 i 个项目在训练集上的预测评分；

$r_{u,i}$——用户 u 对第 i 个项目的实际评分。

MAE 通过计算用户的预测评分和实际评分之间的偏差来度量算法预测的准确性。MAE 值越小，则推荐精确度越高。

3.4.3 结果分析

1. 相似度比较

为了验证本书提出的算法与传统相似度算法在不同相似度公式中的推荐效果，将式（3-2）中的用户兴趣度权重函数引入改进 Pearson 相似度、改进余弦相似度和改进调整余弦相似度计算公式中实现 3 种改进相似度与传统相似度的 MAE 比较，如图 3-2 所示为 3 种改进相似度与传统相似度的 MAE 比较。

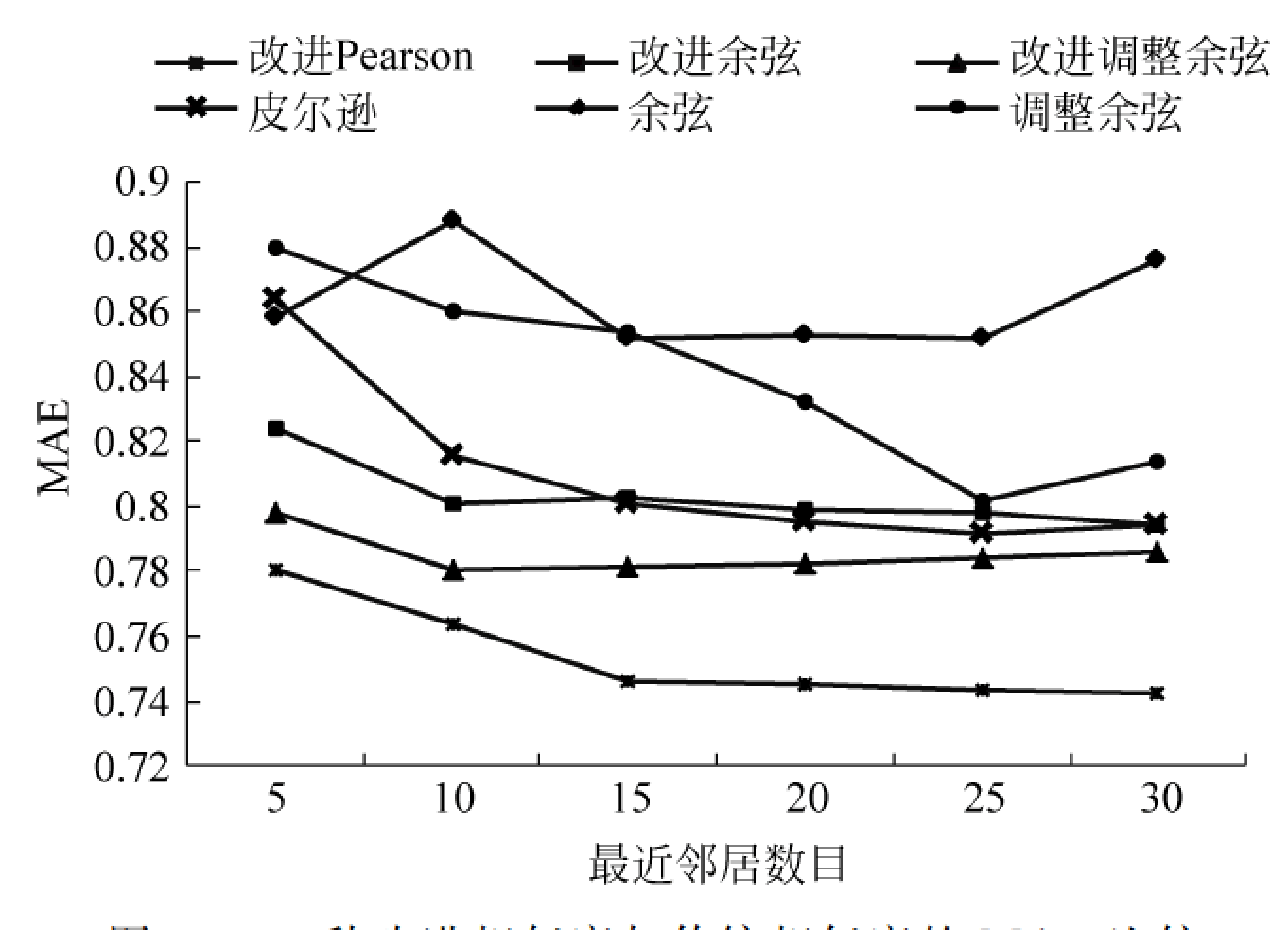


图 3-2　3 种改进相似度与传统相似度的 MAE 比较

分别使用这 3 种计算方式进行 MAE 对比，从图 3-2 中可以看出，不论选用何种相似度计算方式，在任意邻居数 neighbor_num 值下，3 种改进的相似度计算方

法都比原始的计算方法取得更低的 MAE 值，尤其是采用 Pearson 相似度计算相似度时 MAE 值最小，从而也验证了本书选用 Pearson 相似度方法作为相似度计算的依据；另外，改进后算法的 MAE 明显小于传统协同过滤推荐系统，说明基于时间的用户兴趣度权重和项目属性对推荐系统的影响比较大。

2. 平衡因子 γ 对 MAE 的影响

实验中选择最近邻居数 neighbor_num= 25，针对式(3-7)，在其他参数一样的情况下，观察不同平衡因子 γ 值对推荐准确度的影响。如图 3-3 所示为平衡因子 γ 对 MAE 的影响。通过实验得知当 $\gamma=0.15$ 时 MAE 值最小，此时不但考虑了项目属性还考虑了时间对相似度计算的影响。

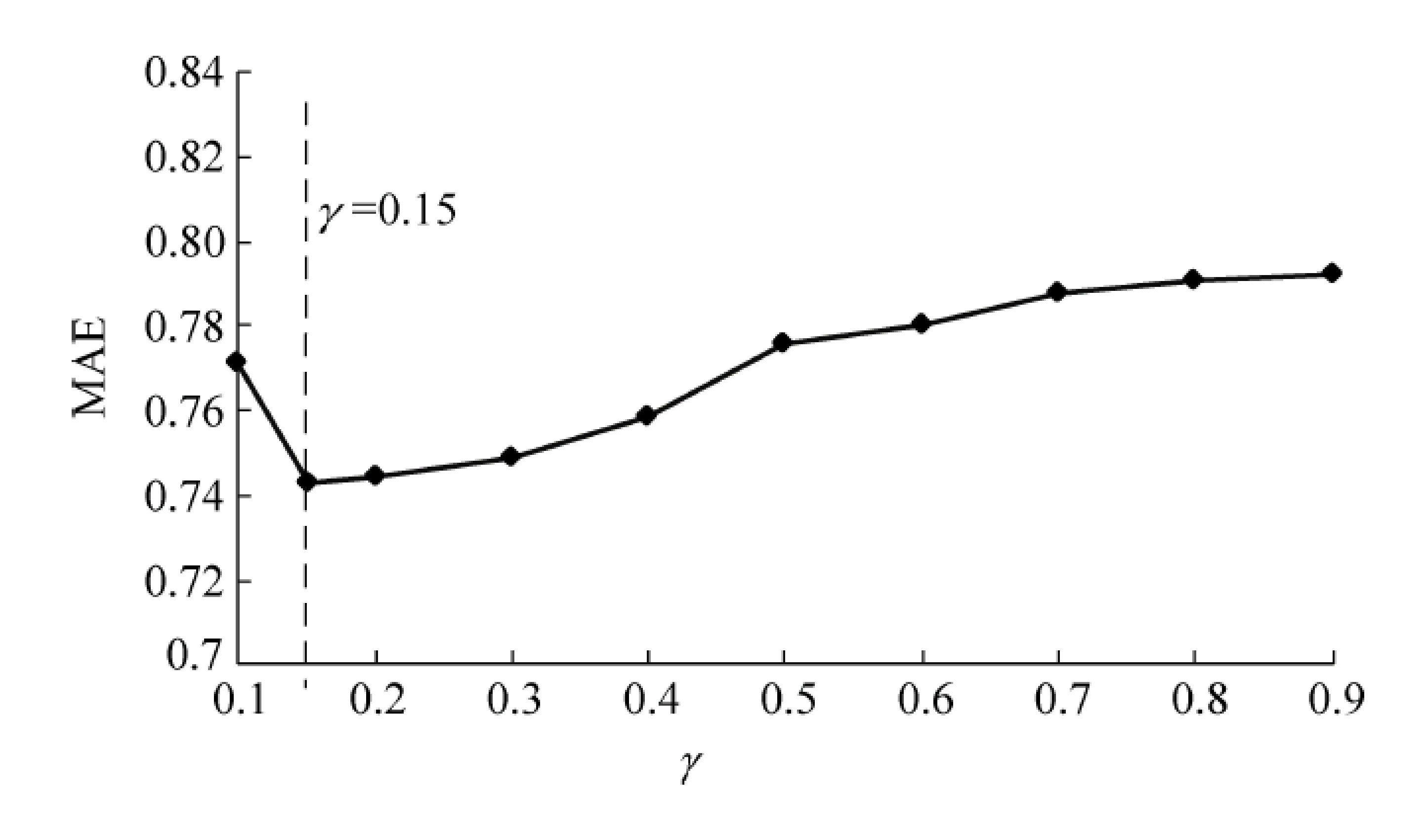


图 3-3　平衡因子 γ 对 MAE 的影响

3. 不同算法之间对比

实验中最近邻居数 neighbor_num 分别取 5、10、15、20、25 和 30，根据实验(2)中设定最优平衡因子 $\gamma=0.15$，本书提出的基于用户兴趣和项目属性协同过滤推荐算法(UIIP-CF)分别与传统基于项目的协同过滤推荐算法(ICF)、文献[9]中提出的改进算法(TDGS-CF)、文献[10]中提出的改进算法(WUCF)进行的 MAE 对比。如图 3-4 所示为 4 种算法的 MAE 对比图。

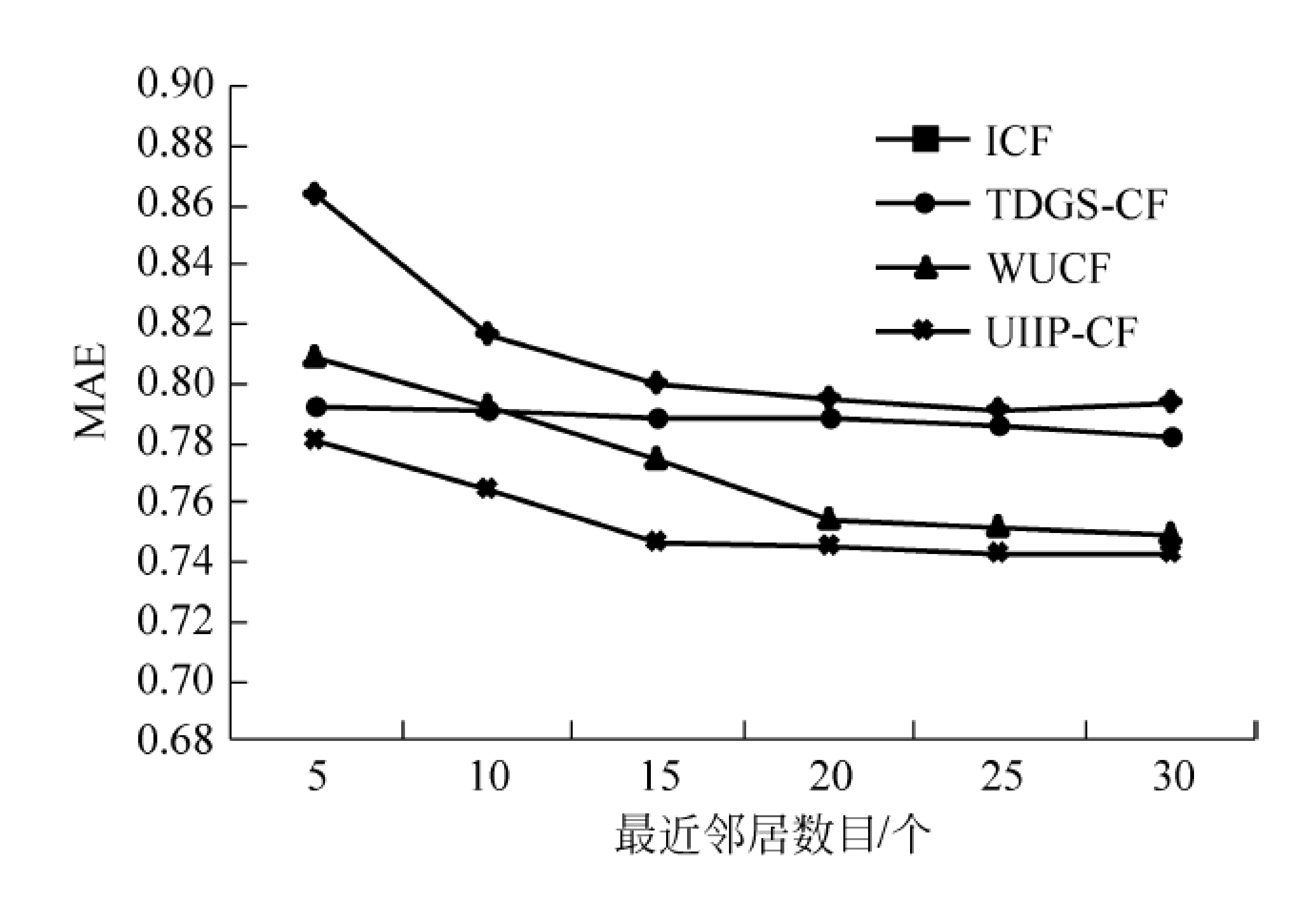


图 3-4　4 种算法的 MAE 对比图

由图 3-4 可以看出，改进算法的 MAE 与传统的基于项目的协同过滤推荐算法相比，MAE 降低了 10%，比 TDGS-CF 算法 MAE 降低了 6%，比 WUCF 算法 MAE 降低了 3%。由于 MAE 值越小则推荐精度越高，UIIP-CF 算法的精确性比上述推荐算法都高，这也正说明了基于时间的用户兴趣权重和项目属性在推荐算法中起着比较关键的作用。

本章小结

本章分析了用户的兴趣随时间的变化而变化的规律，在此基础上提出了一种基于用户兴趣和项目属性的协同过滤推荐算法。实验表明，改进的协同过滤推荐算法不仅有效提高了推荐系统的推荐精度，而且在一定程度上解决了协同过滤推荐系统的用户兴趣漂移问题。改进算法中采用实验法得到最佳平衡因子进行相似度融合以达到解决用户兴趣漂移的目的。

参考文献

[1] 孟祥武，刘树栋，张玉洁. 社会化推荐系统研究[J]. 软件学报，2015，26(6)：1356-1372.

[2] 王国霞，刘贺平，李擎. 基于万有引力的个性化推荐算法[J]. 工程科学学报，2015(2)：255-259.

[3] Adibi P，Ladani BT. A collaborative Filtering Recommender System-based on User's Time Pattern Activity[C]//Information and Knowledge Technology (IKT)，2013 5th Conference on IEEE，Kharagpur，2013：252-257.

[4] Jia C X，Liu R R. Improve the Algorithmic Performance of Collaborative Filtering by Using the Interevent Time Distribution of Human Behaviors[J]. Physica A Statistical Mechanics & Its Applications，2015，436：236-245.

[5] Ładyżyński P，Grzegorzewski P. Vague preferences in recommender systems [J]. Expert Systems with Applications，2015，42(24)：9402-9411.

[6] 郑志高，刘京，王平. 时间加权不确定近邻协同过滤推荐算法[J]. 计算机科学，2014，41(8)：7-12.

[7] 李源鑫，肖如良，陈洪涛. 时间衰减制导的协同过滤相似度计算[J]. 计算机系统应用，2013，22(11)：129-134.

[8] 刘东辉，彭德巍，张晖. 一种基于时间加权和用户特征的协同过滤推荐算法[J]. 武汉理工大学学报，2012，34(05)：144-148.

[9] Chen C，Yin H，Yao J，et al. TeRec：A Temporal Recommender System over Tweet Stream[J]. Proceedings of the Vldb Endowment，2013，6(12)：1254-1257.

[10] Huang Y，Cui B，Zhang W，et al. TencentRec：Real-time Stream Recommendation in Practice[C]//ACM SIGMOD International Conference on Management of Data. ACM，Melbourne，2015：227-238.

[11] Yin H，Cui B，Lu H，et al. A Unified Model for Stable and Temporal Topic Detection

from Social Media Data [C]//2013 IEEE 29th International Conference on Data Engineering (ICDE). IEEE Computer Society, Brisbane, 2013: 661-672.

[12] Yin H, Bin C, Ling C, et al. A Temporal Context-aware Model for User Behavior Modeling in Social Media Systems[C]//Association for Computing Machinery. Special Interest Group on Management of Data. International Conference Proceedings. Association for Computing Machinery, Salt Lake City, 2014: 1543-1554.

[13] Ren Y, Li G, Zhou W. Learning User Preference Patterns for Top-N Recommendations [C]//Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on. IEEE, HongKong, 2012: 137-144.

[14] 吴毅涛,张兴明,王兴茂.基于用户模糊相似度的协同过滤推荐算法[J].通信学报, 2016, 37(1): 198-206.

[15] 荣辉桂,火生旭,胡春华.基于用户相似度的协同过滤推荐算法[J].通信学报, 2014, 35 (2): 16-24.

[16] 朱强,孙玉强.一种基于信任度的协同过滤推荐方法[J].清华大学学报(自然科学版), 2014(3): 360-365.

[17] 焦东俊.基于用户人口统计与专家信任的协同过滤算法[J].计算机工程与科学,2015 (01): 179-183.

[18] 王瑞琴,蒋云良,李一啸,等.一种基于多元社交信任的协同过滤推荐算法[J].计算机研究与发展,2016, 53(6): 1389-1399.

[19] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8): 30-37.

[20] Paterek A. Improving Regularized Singular Value Decomposition for Collaborative Filtering[C]. //Preceedings of KDD Cup and Workshop,California,2007,39-42.

[21] WengJ, Miao C, Goh A. Improving Collaborative Filtering with Trust-based Metrics. [J]. Sac Proceedings of the Acm Symposium on Applied Computing, 2006: 1860-1864.

[22] Moradi P, Ahmadian S. A Reliability-based Recommendation Method to Improve trust-aware Recommender Systems [J]. Expert Systems with Applications, 2015, 42 (21): 7386-7398.

[23] Ghazanfar M A, Prugel-Bennett A. The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations [J]. Informatics, 2013, 37(1): 61-92.

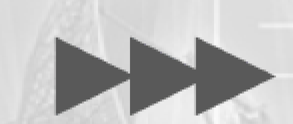
[24] Scott D, Dumais S T, Furnas G W, et al. Indexing by latent Semantic Analysis[J]. Journal of the American Society for Information Science, 1990, 41(6): 391-407.

[25] 夏小伍,王卫平.基于信任模型的协同过滤推荐算法[J].计算机工程, 2011, 21 期(21): 26-28.

[26] Chin W S, Yuan B W, Yang M Y, et al. LIBMF: A Library for Parallel Matrix Factorization in Shared-memory Systems[J]. Technical report, 2015: 32-37.

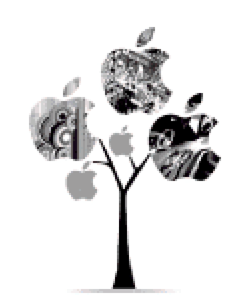
[27] 邓仙荣.基于梯度提升回归算法的 O2O 推荐模型研究[D].合肥:安徽理工大学, 2016.

第三篇　基于矩阵分解的协同过滤推荐算法



矩阵分解模型最早由 Yehuda Koren 于 2008 年提出，主要目的是找到两个低维的矩阵，它们相乘之后得到的矩阵的近似值，与评分矩阵中原有值的位置中的值尽可能接近。近年来，学术界的研究和工业界的应用结果表明，在处理高维数据方面，个性化推荐中使用矩阵分解模型要明显优于传统的基于邻域的协同过滤（又称基于内存或者记忆的协同过滤）方法，如 User-CF、Item-CF 等，这也使得矩阵分解成为目前个性化推荐研究领域中的主流模型。国此，国内外学者尝试采用各种方法来进行对稀疏矩阵进行降维，常用的降维技术有 SVD（Singular Value Decomposition）、矩阵分解、概率矩阵分解、非负矩阵分解以及其改进模型等。

第 4 章 SVD 和信任因子相结合的协同过滤推荐算法



针对推荐系统在数据稀疏情况下推荐质量不高的问题，提出将奇异值分解(Singular Value Decomposition，SVD)技术和信任模型相融合的协同过滤推荐算法。首先运用 SVD 降维技术得到项目的隐式特征空间。然后用改进的余弦相似度计算项目间相似度，根据 k 近邻(k-Nearest Neighbor，k-NN)算法得到第一阶段最近邻居集。接着引入项目信任因子，建立信任模型并融入相似度空间中，进行第二阶段 k 近邻选择，从而完成预测推荐。最后在 MovieLens 数据集与传统的算法进行实验对比，本章提出的算法其标准误差 RMSE(Root Mean Squared Error)的精度提高了 0.53%。

4.1 引言

个性化推荐技术是解决信息过载问题的有效手段。协同过滤是推荐系统中运用最早和最成功的一种推荐技术之一，其原理是采用最近邻居技术，根据用户对项目的偏好，发现用户间或项目间的相关性进行推荐。实际操作过程中，由于用户精力有限，造成大量项目未得到用户的评分，以及随着信息技术的快速发展，用户和项目的数量大幅增加，用户评价过的项目占所有项目的比例很小，因此导致评价矩阵极为稀疏。在这种情况下，计算用户或项目的相似度并不准确，从而造成推荐质量急剧下降。

针对数据稀疏性问题，常采用的处理方法有：

(1) 数据平滑技术：对用户尚未评分的项目进行缺省值填充；运用其他用户对该项目的评分均值来填充缺省值；将用户信息聚类后，以分类用户的评分来填充缺省值。

(2) 数据降维技术：将数据从高维降到低维空间。降维技术主要有主成分分析(Principal Component Analysis，PCA)和奇异值分解(Singular Value Decomposition，SVD)。SVD 是一种有效的数据降维技术，并有着自动抽取矩阵重要特征的能力。文献[6]将 SVD 技术与协同过滤技术结合，此设想在 Netflix Prize 中得到证实并迅速推广。在文献[6]的基础上，加入用户和项目的偏置信息，并提出用户—项目特征函数，提出了 NSVD 和 NSVD2 算法。

传统的协同过滤推荐算法只考虑用户或项目间的相似度这一种因素，而在基于信任因子的协同过滤推荐算法中，用户或项目间的信任度也被作为一个重要因素，并有效提高了算法的推荐精度。文献[7]提出了一种数据挖掘算法对稀疏评分矩阵进行填充；在完整的填充矩阵上计算用户相似性，并引入相似性信任因子，已取得较好的推荐结果。文献[8]提出基于用户间信任因子的协同过滤推荐算法，提高了推荐的精度。

针对以上情况，本章提出将 SVD 技术和信任模型相结合的协同过滤 CFSVD-TF(Algorithm of Collaborative Filtering Based on SVD and Trust Factors)算法。首先运用 SVD 得到项目的隐式特征空间，然后用调整的余弦相似度计算项目间相似度，生成临时邻居集；接着引入信任因子的概念，建立可计算的信任模型并融入到相似度空间中，以做出更高精度推荐预测；最后在 MovieLens 数据集对本章提出的算法进行验证。该算法不仅可以有效地对稀疏矩阵进行降维，而且改变传统的基于项目的协同过滤推荐过程中项目间相似度作为唯一决定预测结果的因素，并证明项目间的信任关系也是影响预测评分结果的重要因素。

4.2 标注和相关工作

4.2.1 标注

为下述表述方便，为本章中使用的标注做统一说明，如表 4-1 所列为用户—项目评分矩阵表，用一个 m 用户和 n 项目的评分矩阵 R 来标示所有用户对所有项目的评分，范围是[1,5]。$R=\{r_{u,i}\}(1\leqslant u\leqslant m,1\leqslant i\leqslant n)$的值由式(4-1)求出。

表 4-1 用户—项目评分矩阵

u	i					
	i_1	i_2	i_3	…	i_{n-1}	i_n
u_1	5	0	3	0	5	0
u_2	0	0	0	0	0	4
u_3	2	0	0	3	0	0
…	0	0	5	1	0	0
u_{m-1}	2	0	0	4	0	0
u_m	0	4	0	0	0	1

$$R=\begin{cases} r_{u,i}, & \text{评分} \\ 0, & \text{未评分} \end{cases} \tag{4-1}$$

式中：$r_{u,i}$——用户 u 对项目 i 的评分，如果用户 u 未对项目 i 评分，那么值为 0。

4.2.2 奇异值分解

在数据挖掘中，SVD 作为一种矩阵分解技术，通过生成一个低秩矩阵近似逼近原始矩阵。给定一个矩阵，则 A 的奇异值分解定义如式(4-1)所示。

$$A = U \times S \times V^{T} \tag{4-2}$$

式中：$U \in R_{m \times m}$；

$V \in R_{n \times n}$；

$S \in R_{m \times n}$。

矩阵 U 和 V 为正交矩阵，且矩阵的列分别为 AA^{T} 和 $A^{T}A$ 的特征向量。r 为矩阵 R 的秩；σ_r 为 R 的奇异值，值为 AA^{T} 或 $A^{T}A$ 的特征值的平方根。因此，这三个矩阵的有效维数分别是 $m \times r$、$r \times r$、$n \times r$。

SVD 可以提供三个矩阵相乘对原始矩阵 A 进行最优的近似。通过将矩阵 S 保留第 k 个最大的奇异值进行简化得到新的对角矩阵，其中 $k < r$。然后通过删除 U 和 V 的相应列，简化后的矩阵 U 和 V 表示为 U_k 和 V_k，如式(4-3)所示。

$$A_{\text{red}} = U_k \times S_k \times V_k^{T} \tag{4-3}$$

这是最接近的 k 位近似原矩阵的酉不变范数。

4.2.3　计算相似度

协同过滤中常采用计算相似度的方法为余弦相似度和改进的余弦相似度。

余弦相似度如式(4-4)所示。

$$\operatorname{sim}(i,j) = \cos(I,J) = \frac{I \times J}{\|I\| \times \|J\|} = \frac{\sum_{s \in S_{ij}} r_{i,s} \times r_{j,s}}{\sqrt{\sum_{s \in S_i} r_{i,s}^2 \sum_{s \in S_i} r_{j,s}^2}} \tag{4-4}$$

式中：$\operatorname{sim}(i,j)$——用户 i,j 的相似度；

I,J——用户 i,j 的评分向量；

S_i,S_j——用户 i,j 的评分项目集；

S_{ij}——用户 i 和 j 共同评分的项目集合；

$r_{i,s},r_{j,s}$——用户 i,j 对项目 s 的评分。

由于基于余弦方法没有考虑不同用户打分的严格程度，可能有的用户偏向于给高分，而有的用户偏向于给低分，该方法通过减去用户打分的平均值消除不同用户打分习惯的影响。改进的余弦相似度如式(4-5)所示。

$$\operatorname{sim}(i,j) = \frac{\sum_{s \in S_{i,j}} (r_{i,s} - \bar{R}_I)(r_{j,s} - \bar{R}_J)}{\sqrt{\sum_{s \in S_i} (r_{i,s} - \bar{R}_I)^2 \sum_{s \in S_j} (r_{j,s} - \bar{R}_J)^2)}} \tag{4-5}$$

式中：$\bar{R}_I$——用户 i 评分的平均值；

$\bar{R}_J$——用户 j 评分的平均值。

4.3　SVD 和信任因子相结合的协同过滤推荐算法

传统的协同过滤算法在生成最近邻居集和产生推荐阶段，仅将用户或项目的相似度作为影响因素。而在现实生活中，人们不仅考虑和自己有相同兴趣的用户

或相似度质的项目，而且项目在所有项目中的口碑好坏也影响着人们的决策，即项目间相似度和被信任程度成为影响预测推荐的重要因素。

4.3.1 项目特征空间

首先将原始评分矩阵 R 中评分值为零的项使用相关列的平均值来替代，然后将矩阵的每行规范化到相同的长度用来替代 $R_{u,i}$，其中 R_u 是相关列的项目的平均评分值。最后运用 SVD 分解技术得到项目隐式特征空间。

4.3.2 两阶段 k 近邻选择

k-NN 分类算法是数据挖掘分类技术之一。所谓 k 近邻算法，即是给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近的 k 个实例（也就是上面所说的 k 个邻居），这 k 个实例的多数属于某个类，就把该输入实例分类到这个类中。本章提出的算法采用两个阶段运用 k-NN 算法来寻找项目的最近邻居，第一阶段是在项目特征空间下，采用改进的夹角余弦方法计算得到项目相似度矩阵，在此，对每个项目取 k（$k=10,20,30$）个邻居，得到项目邻居群 Q_i。第二阶段是在将项目信任因子融入相似度空间后，再次寻找最近邻居。

4.3.3 信任因子

在协同过滤系统里，项目在整个用户—项目评分矩阵中被信任的程度称为信任因子。信任因子包括全局信任（Global Trust）和局部信任（Local Trust）两部分内容。

全局信任：单个项目在全部项目中的信誉。用 T_i 表示项目 i 全局信任，且 $0\leqslant T_i\leqslant 1$，如式（4-6）所示。

$$T_i=\frac{2[1-1/\ln(f_i+2)]\times[1-1/\ln(q_i+3)]}{2-1/[\ln(f_i+2)-1/\ln(q_i+3)]} \tag{4-6}$$

式中：f_i——项目被用户评价的次数；

q_i——该项目被其他项目作为邻居的次数。

局部信任：每两个项目间的信任值，若两个项目间相似度越高，则两个项目的信任程度就越高。因此，局部信任由全局信任和相似度两个因素决定。结合式（4-5）、式（4-6），局部信任如式（4-7）所示。

$$T_b(P_a)=\frac{2\times \mathrm{sim}(a,b)\times T_a}{\mathrm{sim}(a,b)+T_a} \tag{4-7}$$

式中：$T_b(P_a)$——项目 b 对项目 a 的局部信任；

$\mathrm{sim}(a,b)$——项目 a 和项目 b 的相似度；

T_a——项目 a 的全局信任。

注意，项目 a 与项目 b 之间和项目 b 与项目 a 的相似度是相等的，即 $\mathrm{sim}(a,b)=\mathrm{sim}(b,a)$，而项目 a 与项目 b 之间和项目 b 与项目 a 之间的局部信任度是不等的，

即 $T_a(P_b) \neq T_b(P_a)$。

4.3.4　预测评分

预测用户 u 对项目 i 的评分如式(4-8)所示。

$$\mathrm{pr}_{u,i} = \frac{\sum_{k=1}^{l} T_i(P_k) \times (rr_{ui} + \overline{r_u})}{\sum_{k=1}^{l} |\ T_i(P_k)\ |} \tag{4-8}$$

式中：l——根据 k-NN 算法得到的项目 i 的邻居数；

rr_{ui}——在通过 SVD 降维后空间 A_{red} 下，用户 u 对项目 i 的评分值；

$\overline{r_u}$——用户 u 的评分均值。

4.3.5　算法

算法基本思想：首先利用奇异值分解得到项目特征空间，利用改进的余弦相似度计算项目间相似度，然后根据 k-NN 算法得到临时邻居集，在此基础上，引入项目的信任因子。SVD 和信任因子相结合的协同过滤推荐算法(CFSVD-TF 算法)流程如算法 4-1 所示。

算法 4-1　CFSVD-TF 算法

输入：评分矩阵 R。

输出：预测矩阵 R_{pred}。

算法的基本流程：

步骤 1：填充原始矩阵 R 并且规范化得到 R_{norm}。使用奇异值分解方法分解 R_{norm} 得到矩阵 U,S,V。

步骤 2：将矩阵 S 进行维数简化到 k 维，得到矩阵 $S_k(k<r, \mathrm{rank}(R_{\mathrm{norm}})=r)$。相应地，将矩阵 U,V 进行化简得到矩阵 U_k,V_k。$R_{\mathrm{red}}=U_k \times S_k \times V_k^{\mathrm{T}}$。计算 S_k 的平方根得到矩阵 $\sqrt{S_k}$，得到项目的隐式特征空间 $\sqrt{S_k} \times V_k^{\mathrm{T}}$。

步骤 3：在项目特征空间下，使用式(4-3)计算项目 i 和项目 j 的相似度 $\mathrm{sim}(i,j)$。

步骤 4：根据 k-NN 算法，设定 $k(k=20)$个邻居，得到项目邻居群 Q_i，由项目邻居群 Q_i 得到 q_i，遍历原始矩阵 R 得到 f_i。

步骤 5：由式(4-6)计算项目的全局信任 T_i，并将 T_i 填充为矩阵，使用式(4-7)计算项目间局部信任。

步骤 6：根据 k-NN 算法，得到项目的最近邻居集。

步骤 7：由式(4-8)进行预测评分。

步骤 8：根据 Top-N 方法，将预测评分最高的 N 个项目推荐给相应的用户。

算法结束

CFSVD-TF 算法利用用户和项目之间的潜在关系达到对维空间的简化，使用奇异值分解来增加数据的密度，将项目的信任因子作为权重加入到相似度计算中，这样不仅能克服数据稀疏性问题，还能避免过于强调相似度的作用，提高了推荐的精度。

4.4 实验结果与分析

4.4.1 数据集和实验环境

本章实验采用美国 Minnesota 大学的 MovieLens100k 数据集。计算稀疏等级值如式(4-9)所示。

$$S_{lv}=1-\frac{N_{ui}}{m\times n} \tag{4-9}$$

式中：S_{lv}——稀疏等级值；

N_{ui}——数据集中用户对项目的评分个数；

m——用户个数；

n——项目个数。

实例说明：该数据集包括 943 位用户对 1682 部电影的 100 000 个评分，评分采用 5 分制，取 1～5 的整数，1 分表示用户认为该部影片不好看，5 分表示用户认为该影片非常好看，打分越高说明用户越喜欢该电影。

数据中已经划分成 5 个数据集(u_1～u_5)，并随机抽取 80%的数据作为训练集，20%的数据作为测试集，采用 5 折交叉法进行验证。

实验采用的 PC 配置为双核 3.50GHz、内存为 4GB，操作系统为 Windows 7。在 MATLAB R2014a 平台上实现了基于项目的协同过滤推荐算法、基于 SVD 的协同过滤推荐算法和基于 CFSVD-TF 算法。

4.4.2 评价标准

本章实验采用 RMSE 作为评价标准，RMSE 通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性。此度量方法直观、严谨，是最常用的一种推荐质量度量方法。RMSE 值越小，表示推荐质量越高。RMSE 计算如式(4-10)所示。

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{N}(P_i-r_i)^2}{N}} \tag{4-10}$$

式中：$\{P_1,P_2,\cdots,P_N\}$——算法对 N 个项目预测的评分集合；

$\{r_1,r_2\cdots,r_N,\}$——实际评分集合。

4.4.3 实验结果分析

1. 相似度的分布与选取

为了得到较佳的实验结果，本章分别用 4.2 节中余弦相似度和改进的余弦相似度方法计算项目之间的相似度，两种相似度的分布如图 4-1 所示。从图 4-1 可

以看出,余弦相似度分布较为均匀,改进的余弦相似度分布更具个性化。在[0,1]范围内,运用余弦相似度得到的相似度值分布在[0.0,0.6],平均达到89.10%,分布过于分散。而改进的余弦相似度得到的相似度值主要分布在[0.0,0.4],平均达到80.15%,因为通过项目的评分值减去用户评分的平均值,均衡了用户的评分尺度不一问题,更真实反映出项目的差异特征,即用户的个性化选择。所以,根据改进的余弦相似度计算方法可以得到较高质量的推荐。基于以上分析,本章采用改进的余弦相似度计算方法进行度量。

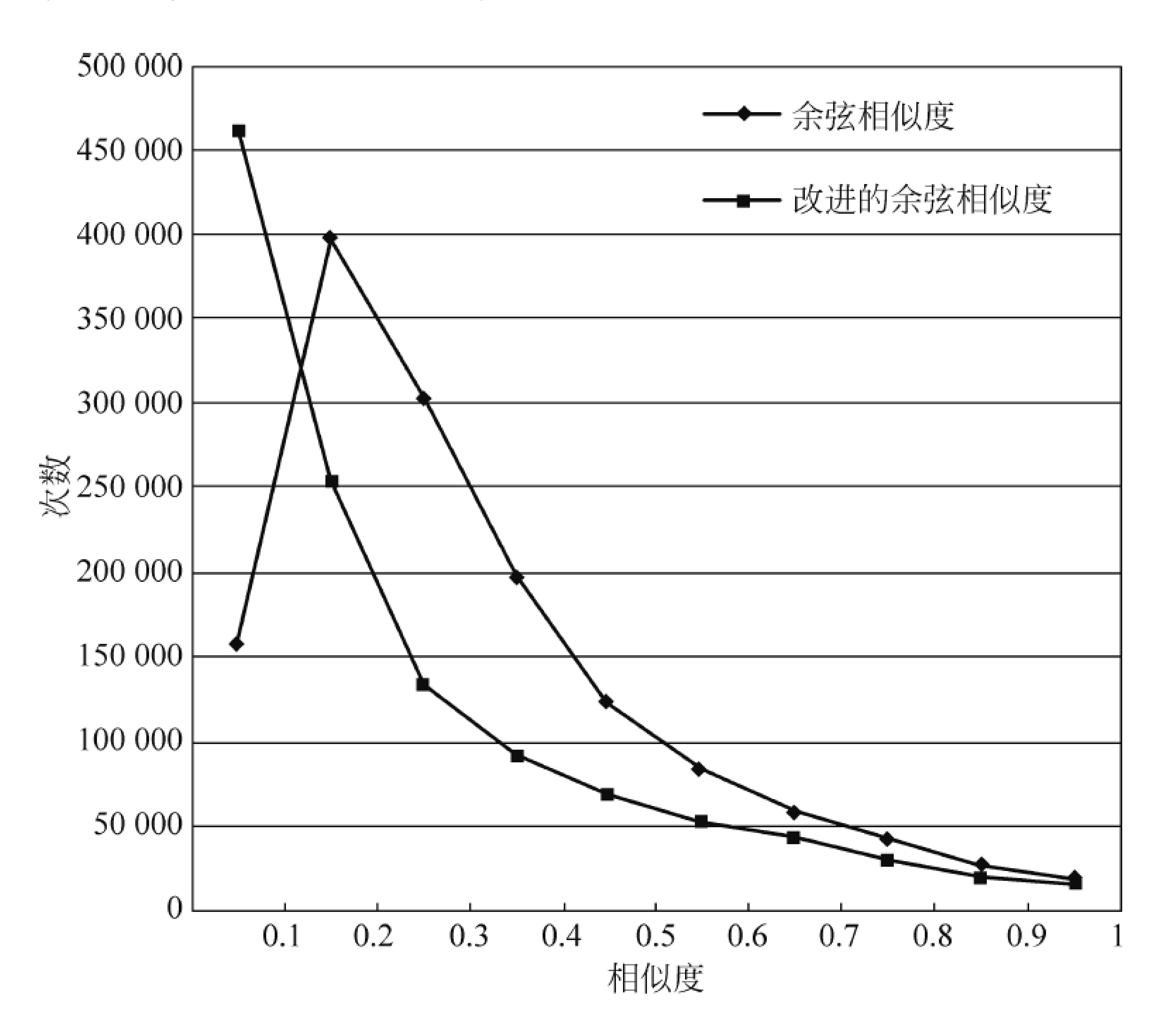


图 4-1　两种相似度的分布

2. 全局信任因子的分布与分析

图 4-2 所示为信任因子 f_i 的分布,在 MovieLens_100K 数据集中,项目被用户评价的次数在[0,500]范围内,其中,67.84%的项目被评价次数在[0,50]区间中,剩余的项目在其他区间都有分布,说明信任因子 f_i 能表示出单个项目在全体项目中个性化的特质。

图 4-3 所示为信任因子 q_i 的分布,在第一阶段寻找最近邻居时,活动的项目作为其他项目的邻居次数分布在[0,800]范围内。在[0,50]区间内,当 K 取 10,20,30 时,项目数目依次为 1603,1522,1469。随着 K 值的增大,信任因子 q_i 的分布变得均匀。根据实验的最终结果分析得出,在 $K=20$ 时,RMSE 得到最优。当 $K=20$ 时,信任因子 q_i 的分布范围在[0,750],将近 10%的项目分布在[50,750]的不同区间中,说明信任因子 q_i 能够作为全局信任的重要因素。

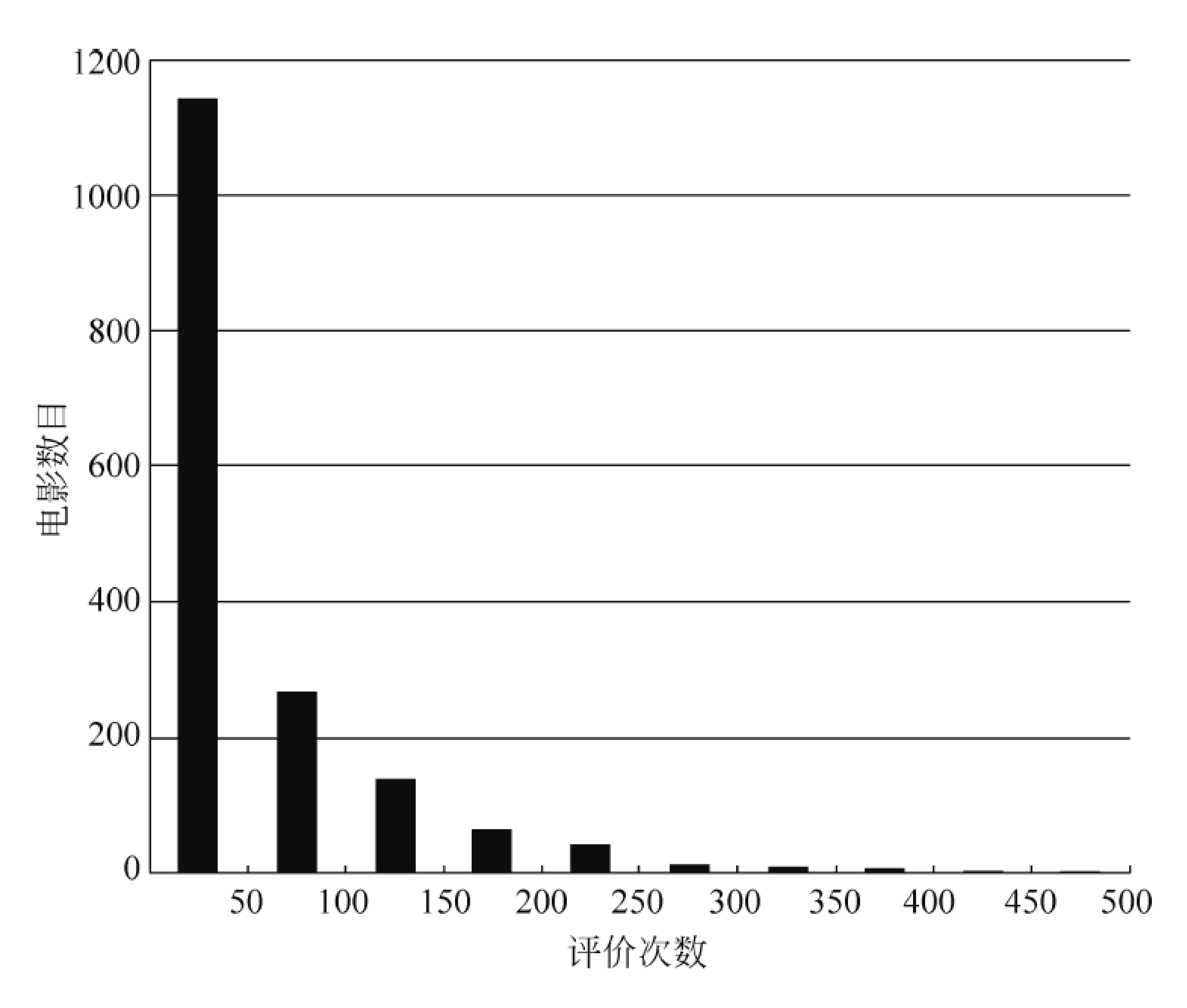


图 4-2 信任因子 f_i 的分布

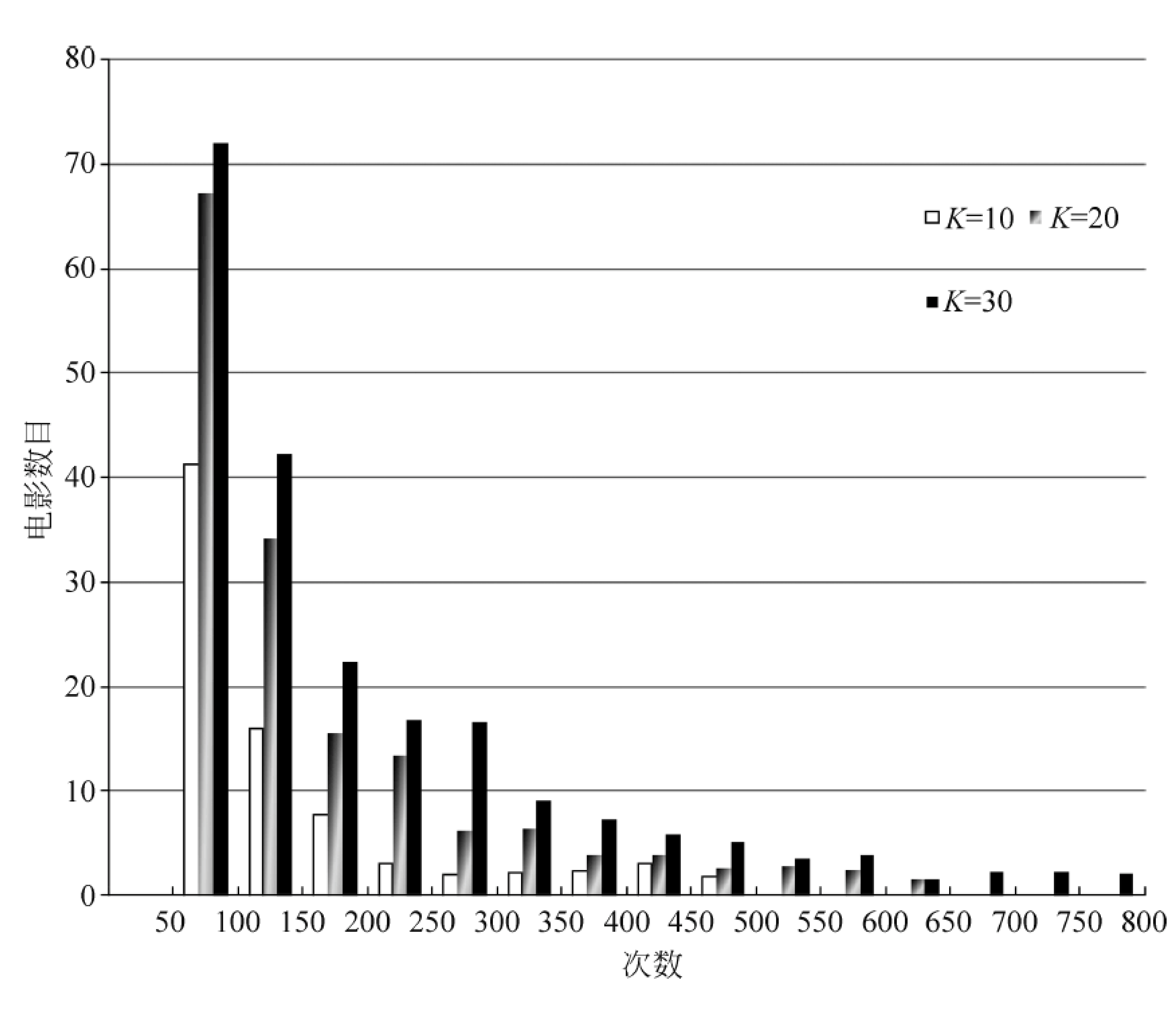


图 4-3 信任因子 q_i 的分布

3. 信任的分布与分析

图 4-4 所示为信任度分布，显示了在不同邻居数情况下信任值的分布情况。可以看出，同相似度分布图比较，信任值分布较为均匀，除了信任值区间[0,0.1]和

[0.2,0.3],其他区间都有分布。其中,[0.4,0.8]区间分布比例最高,占总分布的88.88%。在推荐系统中,通过信任算法,每个项目都得到一个信任值,90%的项目信任值分布在[0.4,0.8],而只有 13.84%项目间相似度值分布在[0.4,0.8]。

因此,信任因子和相似度是完全不同的两个因素,把信任因子引入协同过滤推荐算法是可行的。

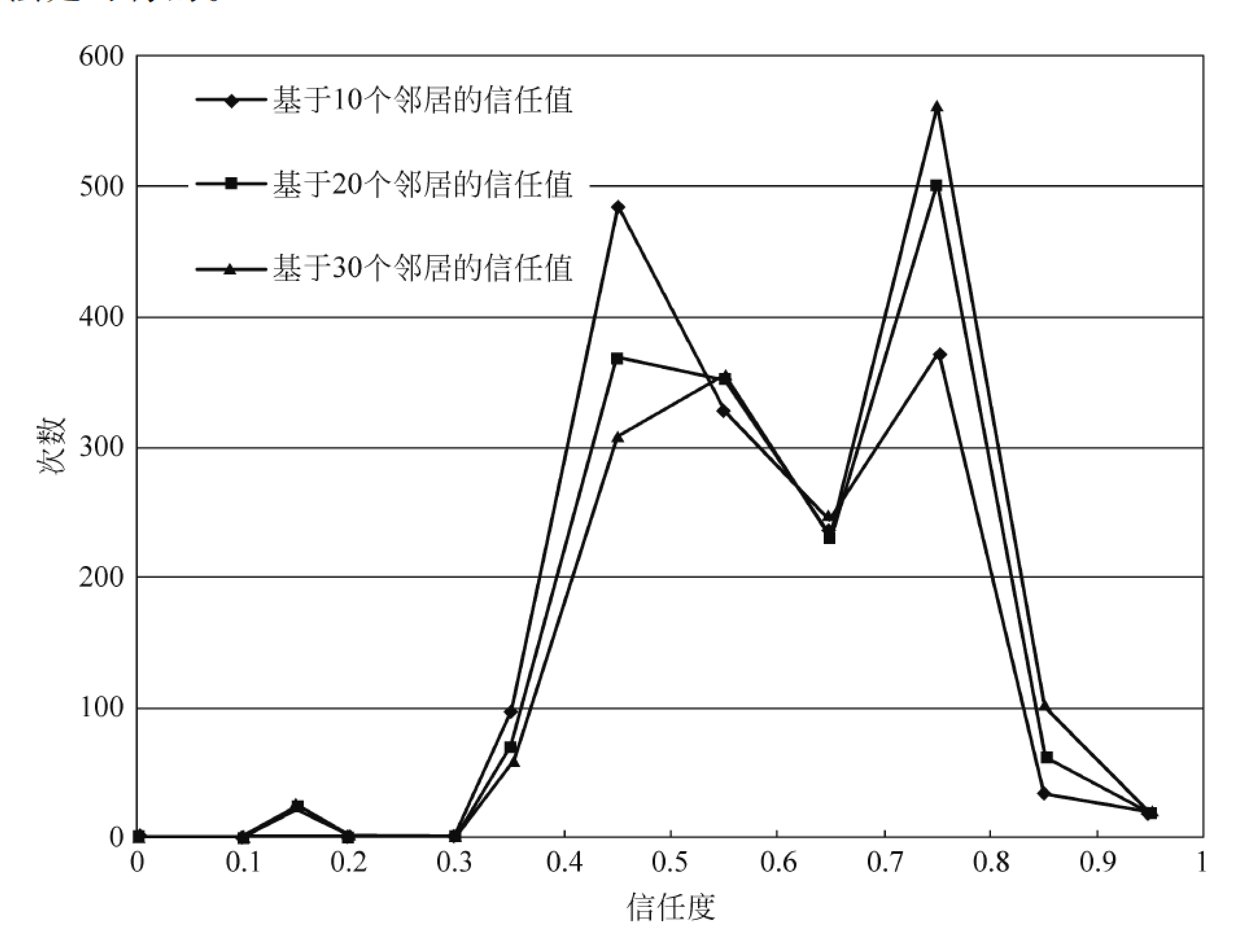


图 4-4　信任度的分布

4. 实验结果及分析

图 4-5 所示为不同推荐策略的 RMSE。可以看出,三种算法中 RMSE 值随着邻居数目的变化而发生变化,相比传统的基于项目的协同过滤(Item-based Collaborative Filtering,ICF)算法和传统的基于 SVD 的协同过滤(Collaborative filtering algorithm based on SVD, SVD-CF)算法,本章提出的 CFSVD-TF 算法在邻居数小于或等于 10 个时,RMSE 值呈现指数式下降;当邻居数大于 10 个时,变化趋稳;当邻居数等于 12 时,推荐性能达到最好,RMSE=0.9762,比 SVD-CF 精度提高了 0.53%。随后,随着邻居数目的增加 RMSE 值又开始反弹,说明邻居数目对于算法的影响较大。CFSVD-TF 算法和 SVD-CF 算法相对于 ICF 算法,RMSE 优化更明显。CFSVD-TF 算法相比 SVD-CF 算法,RMSE 值一直处于下降状态并且推荐精度更优,说明本章提出的算法是有效可行的,不仅能有效增大数据密度,而且有效提高了协同过滤推荐算法的预测精度。

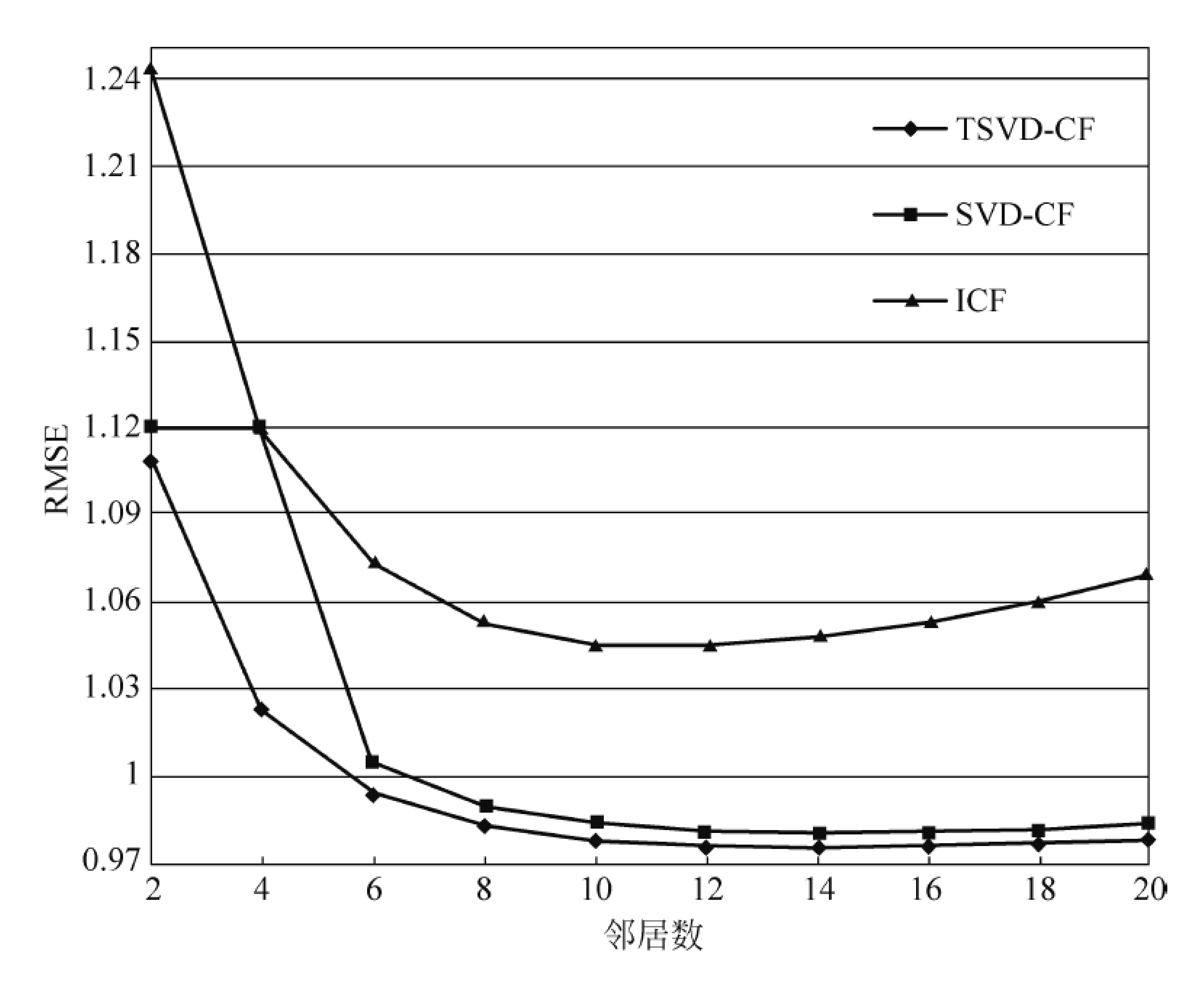


图 4-5 不同推荐策略的 RMSE

本章小结

本章在分析传统的基于项目的协同过滤推荐算法及数据稀疏性问题的基础上，提出基于 CFSVD-TF 算法。首先采用 SVD 方法对数据特征进一步挖掘，缓解了数据稀疏性问题；然后引入项目信任因子，改变了项目间相似度作为唯一决定预测结果的情况。实验结果证明，本章算法能够缓解数据稀疏性带来的推荐精度不高的问题，能够满足此应用领域的要求。在下一步的研究工作中，将会采用带反馈的 SVD 结合更高维的特征向量，进一步研究协同过滤推荐算法中，根据项目间的相关规律所能反映出的用户个性化，进一步提高算法的推荐精度。

参考文献

[1] Leng Ya-jun, Chen Qing, Liang Chang-yong. Survey of Recommendation Based on Collaborative Filtering [J]. Pattern Recognition & Artificial Intelligence, 2014 (8): 720-734.

[2] Moradi P, Ahmadian S. A Reliability-based Recommendation Method to Improve Trust-aware Recommender Systems [J]. Expert Systems with Applications, 2015, 42(21): 7386-7398.

[3] Deng Ai-lin, Zhu Yang-yong, Shi Bo-le. Survey of *R* Recommendation Based on Collaborative Filtering [J]. Journal of Software, 2003, 14(9): 1621-1628.

[4] Wang Q M, Liu X, Zhu R, et al. A New Personalized Recommendation Algorithm of Combining Content-based and Collaborative Filters [J]. Computer &Modernization, 2013, 1(8): 64-67.

[5] Kong Wei-liang. Research on the Key Problems of Collaborative Filtering Recommender System[D]. Central China Normal University, 2013.

[6] Ma Chihchao. A Guide to Singular Value Decomposition for Collaboratative Filtering [J]. Csientuedutw, 2008.

[7] Paterek A. Improving Regularized Singular Value Decomposition for Collaborative Filtering [C]. //Preceedings of KDD Cup and Workshop,California,2007,39-42.

[8] Tsai C F, Hung C. Cluster Ensembles in Collaborative Recommendation [J]. Applied Soft Computing, 2012, 12(4): 1417-1425.

[9] Guo Yan-hong, Deng Gui-shi, Luo Chun-yu. Collaborative Filtering Recommendation Algorithm Based on Factor of Trust [J]. Computer Engineering, 2008, 34(20): 1-3.

[10] Ghazanfar M A, Prugel-Bennett A. The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations [J]. Informatics, 2013, 37(1): 61-92.

[11] Zhou X, He J, Huang G, et al. SVD-based Incremental Approaches for Recommender Systems [J]. Journal of Computer & System Sciences,2015,81(4): 717-733.

[12] Sarwar B, Karypis G, Konstan J, et al. Item-based Collaborative Filtering Recommendation Algorithms[C]. //Proceedings of the 10th International Conference on World Wide Web. ACM, 2001, 285-295.

[13] Sang H C, Cho Y H. An Utility Range-based Similar Product Recommendation Algorithm for Collaborative Companies[J]. Expert Systems with Applications, 2004, 27(4): 549-557.

[14] Hwang W S, Lee H J, Kim S W, et al. Efficient Recommendation Methods Using Category Experts for a Large Dataset[J]. Information Fusion, 2016, 28(C): 75-82.

[15] Cho J, Kwon K, Park Y. Collaborative Filtering Using Dual Information Sources[J]. IEEE Intelligent Systems, 2007, 22(3): 30-38.

[16] Jiang W, Yang L. Research of Improved Recommendation Algorithm Based on Collaborative Filtering and Content Prediction[C]. //2016, 26(2): 2330-2338.

[17] Wang Q M, Liu X, Zhu R, et al. A New Personalized Recommendation Algorithm of Combining Content-based and Collaborative Filters [J]. Computer & Modernization, 2013, 1(8): 64-67.

[18] Marinho L B, Hotho A, Jäschke R, et al. Baseline Techniques[M]. Springer US, 2012.

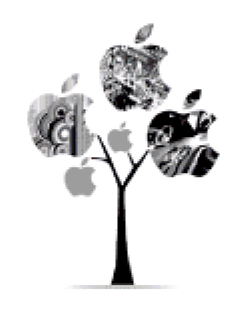
[19] 彭飞,邓浩江,刘磊. 加入用户评分偏置的推荐系统排名模型[J]. 西安交通大学学报, 2012, 46(6): 74-78.

[20] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems [J]. Computer,2009, 42(8): 30-37.

[21] Lin C J. Projected Gradient Methods for Nonnegative Matrix Factorization[J]. Neural Computation, 2007, 19(10): 2756.

[22] Aharon M, Elad M, Bruckstein A.-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation [J]. IEEE Transactions on Signal Processing, 2006, 54(11): 4311-4322.

第 5 章



相似度填充的概率矩阵分解的协同过滤推荐算法

针对基于用户的协同过滤推荐算法中用户间交叉评分项较少的情况，提出一种改进相似度与社交网络中信任因子相结合进行计算的新方法。首先利用原始稀疏评分矩阵计算不同用户间评分项目的并集和交集，以获取一方评价而另一方缺失评价的项目评分集合。然后通过概率矩阵分解技术进行降维从而获得近似评分矩阵，用近似评分矩阵动态填充上述项目评分集合，以替代用户共同评分项的方式来计算用户间相似度；同时对于部分填充中存在误差的项目通过信任度因子来动态调整，以其获得更符合实际环境下的相似度。通过不同形式的实验对比结果，表明本文提出的算法在预测精度方面显著提高。

5.1 引言

互联网由 Web 2.0 时代进入 Web 3.0 时代，用户逐渐由信息消费者扩展到信息生产者与消费者。随着用户参与信息生产，网络信息规模呈爆炸式增长。海量信息为信息检索提供了可能的同时也导致了信息过载。为了缓和这种矛盾，帮助用户在海量数据中准确快速找到其感兴趣的信息，推荐系统应运而生。由于协同过滤能够处理电影、音乐、商品推荐等难以进行文本描述的项目，因而广泛应用于电子商务行业。虽然协同过滤具有显著优秀性能，但仍然面临很多问题，其中之一是实际应用中数据往往极度稀疏。以电子商务为例，在电子商务系统中用户购买的商品通常不足网站商品总数的 1%，用户只对极少数商品进行评分。常规的相似度计算方法仅使用共同评价项目，用户间具有隐式相似度，但由于没有共同评分项而无法计算其相似度。

国内外学者提出采用将降维技术来缓解推荐算法中的数据稀疏问题。Sarwar 等人首先提出采用奇异值分解(Singular Value Decomposition，SVD)，以矩阵分解角度实现降维，提取隐因子信息。Ruslan Salakhutdinov 等人提出概率矩阵分解(Probabilistic Matrix Factorization，PMF)技术，给予 SVD 概率解释并加以正则项避免过拟合。我国台湾学者林智仁提出了支持向量机的研究对降维技术进行改

进。降维技术在保留大部分数据信息的情况下减少数据维数，虽然取得了一定成果，但不可避免地损失一部分有用信息。为提高数据利用率，研究人员提出了改进相似度计算方法。J. Bobadilla 等人提出利用均值填补缺失信息以充分挖掘用户特征信息。孙小华等人综合基于 SVD 的协同过滤推荐算法和基于 k 近邻的协同过滤推荐算法两者的优势，提出了 Pear_Afrer_SVD 算法。该算法先使用 SVD 技术对原始评分矩阵 R 进行分解，再通过分解矩阵逆向求解近似评分矩阵，之后利用填充后的近似评分矩阵进行用户相似度计算，最后采用 k 近邻算法选择目标用户的邻居，并通过邻居做出推荐预测。郝立燕等人提出用 SOFT_IMPUTE 算法补全稀疏的评分矩阵结合相似度因子与 k 近邻算法做出推荐预测，通过补全的评分矩阵加以信任因子限制得到 WCF-SOFT 算法。基于填补的相似度计算方法不可避免地会使预测评分参与计算，影响原始用户特征信息。杨兴耀等人提出基于信任模型填充的协同过滤推荐模型(Collaborative Filtering a Recommendation Model Based on Trust Model Filling，CFTM)，该方法通过分析日常人类行为习惯，利用评分矩阵采样建立信任模型对用户相似度进行填充，然而单纯信任因子无法充分挖掘用户特征信息。

基于上述问题，提出一种基于用户和相似度填充的协同过滤推荐算法(User-based Collaborative Filtering Algorithm Integrating a Part of Filling and Confidence Factor，CF-PFCF)，通过部分填充评分矩阵，用户所有的评价行为可以被充分挖掘，针对稀疏矩阵，有效避免了大量用户间没有共同评分项的问题。然后引入用户信任因子，有效衡量每位用户评价信息的可信性和可靠性，避免用户的恶意评分行为。实验结果表明，CF-PFCF 算法与传统相似度填充算法和信任因子填充算法相比，具有更高的预测精度。

5.2 相关工作

5.2.1 协同过滤推荐算法

经过多年的研究与实验，协同过滤模型已经成为个性化推荐系统中应用最广泛的模型。典型的协同过滤可以分为基于物品的协同过滤和基于用户的协同过滤，本研究使用后者实现。协同过滤需要用户对项目的兴趣度，该信息通常以用户—评分矩阵的形式表示。例如，一个包含 n 个用户，m 个项目的用户—评分矩阵。

$$R_{n\times m}=\begin{bmatrix} r_{1,1} & \cdots & r_{1,k} & \cdots & r_{1,m} \\ \vdots & & \vdots & & \vdots \\ r_{2,1} & \cdots & r_{2,k} & \cdots & r_{2,m} \\ \vdots & & \vdots & & \vdots \\ r_{n,1} & \cdots & r_{n,k} & \cdots & r_{n,m} \end{bmatrix}$$

矩阵中每一行 r_i 表示用户 i 评价电影的集合，所有用户集合用 U 表示；每一列 r_j 表示评价电影 j 的用户集合，所有电影集合用 V 表示；每一个元素 $r_{i,j}$ 表示用户 i 对电影 j 的评分。

1．计算用户似度

利用评分矩阵计算用户 u,v 的相似度 $\mathrm{sim}(u,v)$，其中 $u,v\in U$。针对不同数据集，不同相似度计算方法体现出不同效果，采用如式(5-1)Pearson 相关系数计算用户相似度。

$$\mathrm{sim}(i,j)=\frac{\sum_{v\in V}(R_{i,v}-\bar{R}_i)(R_{j,v}-\bar{R}_j)}{\sqrt{\sum_{v\in V}(R_{i,v}-\bar{R}_i)^2}\sqrt{\sum_{v\in V}(R_{j,v}-\bar{R}_j)^2}} \tag{5-1}$$

式中：$\bar{R}_i$——用户 i 评价所有电影的平均评分；

$\bar{R}_j$——用户 j 评价所有电影的平均评分。

Pearson 相关系数用来衡量两个向量之间的线性相关程度。当两个向量线性关系增强时，Pearson 相关系数趋于 1 或－1。

2．确定邻居集合

为目标用户 u_0 确定邻居集合 U_{u_0} 时，通常采用 k 邻近算法进行计算。例如，预测 $r_{u_0,j}$，首先找出对电影 j 评价过的用户集合 U_j，计算 u_0 与集合 U_j 中元素的相似度并进行降序排序。挑选 U_j 中前 k 个元素构成 u_0 的邻居集合 U_{u_0}。

3．预测评分

预测目标用户 u_0 对电影 j 的评分公式如式(5-2)所示。

$$r_{u_0,j}=\bar{r}_{u_0}+\frac{\sum_{u\in U_{u_0}}(r_{u,j}-\bar{r}_u)\times \mathrm{sim}(u_0,u)}{\sum_{u\in U_{u_0}}|\mathrm{sim}(u_0,u)|} \tag{5-2}$$

式中：$\bar{r}_{u_0}$——用户 u 的平均评分值；

U_{u_0}——u 的邻居集合；

$r_{u,j}$——邻居 u 对电影 j 的评分；

$\bar{r}_u$——邻居 u 的平均评分值；

$\mathrm{sim}(u_0,u)$——u_0 与邻居 u 的相似度。

5.2.2 概率矩阵分解技术

PMF 是现代推荐系统的基础算法之一，核心思想是：假设用户与电影间的关系可以由少数几个因素的线性组合决定。用矩阵的角度来描述，评分矩阵 R 可以分解为两个低维矩阵的乘积 $R=U^{\mathrm{T}}V$，其中矩阵 U 为 $k\times n$ 阶矩阵，描述用户的 k 个属性；矩阵 V 为 $k\times m$ 阶矩阵，描述电影的 k 个属性。根据秩的性质，k 不得大于矩阵 R 的秩。通过分解出的用户特征矩阵 U 和电影特征矩阵 V，逆向可求得近似评分矩阵$\hat{R}$。如图 5-1 为 PMF 的概率图模型。

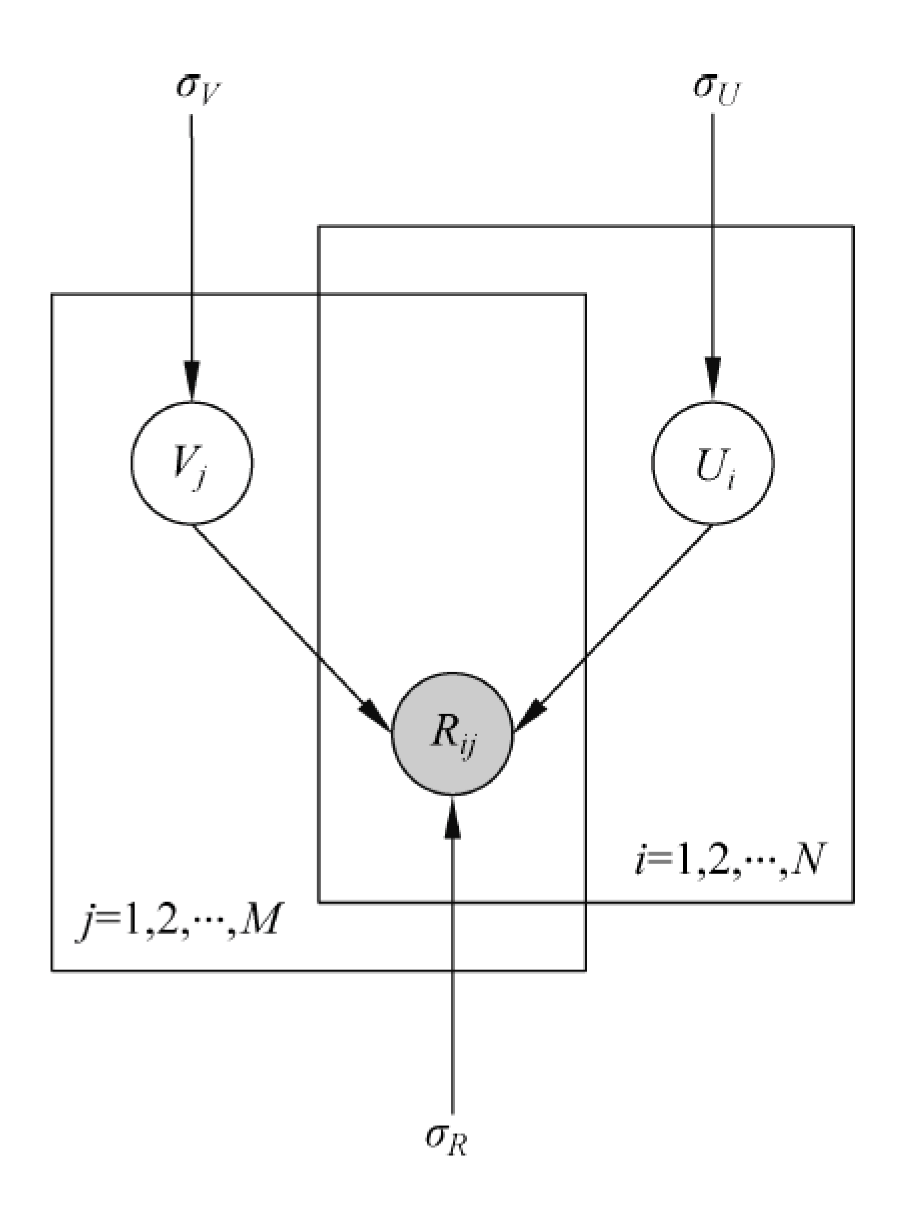


图 5-1　PMF 概率图模型

PMF 基于以下两个假设：

(1) 观测评分矩阵 R 与近似评分矩阵 $\hat{R}$ 的差(观测噪声)符合高斯分布。由贝叶斯观点，完整观测矩阵的概率密度函数如式(5-3)所示。

$$p(R|U,V)=N(\hat{R},\sigma^2)=N(U^{\mathrm{T}}V,\sigma^2) \tag{5-3}$$

式中：σ——观测噪声的方差，需人工设定(设定为 0.1)。

(2) 用户特征属性矩阵 U 和电影特征矩阵 V 符合高斯分布。其概率密度函数如式(5-4)所示。

$$p(U)=N(0,\sigma_U^2),\quad p(V)=N(0,\sigma_V^2) \tag{5-4}$$

式中：σ_U,σ_V——先验噪声的方差，需人工设定(为 0.1)。

综合以上两个概率密度函数，经贝叶斯后验概率如式(5-5)所示。

$$p(U,V\mid R)=\frac{p(U,V,R)}{p(R)}\propto p(U,V,R)=p(R\mid U,V)p(U)p(V) \tag{5-5}$$

对上述预测公式取对数，可以得到式(5-6)。

$$\begin{aligned}\ln p(U,V\mid R,\sigma^2,\sigma_V^2,\sigma_U^2)=&-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}(R_{ij}-U_i^{\mathrm{T}}V_j)^2-\frac{1}{2\sigma_U^2}\sum_{i=1}^{N}U_i^{\mathrm{T}}U_i-\\&\frac{1}{2\sigma_V^2}\sum_{j=1}^{M}V_j^{\mathrm{T}}V_j-\frac{1}{2}\Big(\Big(\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\Big)\ln\sigma^2+\\&ND\ln\sigma_U^2+MD\ln\sigma_V^2\Big)+C\end{aligned} \tag{5-6}$$

式中：C——一个不依赖于参数的常数。

最大化 U 和 V 的后验概率如式(5-7)所示。

$$E(U,V)=\frac{1}{2}\sum_{ij}I_{ij}(R_{ij}-U_i^{\mathrm{T}}V_j)^2+\frac{\lambda_U}{2}\sum_{i}U_i^{\mathrm{T}}U_i+\frac{\lambda_V}{2}\sum_{j}V_j^{\mathrm{T}}V_j \tag{5-7}$$

式中：$R_{i,j}$——用户 i 对电影 j 的评分；

U_i——用户特征矩阵中第 i 行行向量；

V_j——电影特征矩阵中第 j 行行向量；

λ_U 和 λ_V——正则化参数，需人工设定(设为 0.01)。

对代价函数式(5-7)进行梯度下降，即可得到用户特征矩阵 U 和电影特征矩阵 V，进而通过公式$\hat{R}=U^{\mathrm{T}}V$ 获得近似评分矩阵$\hat{R}$。

5.3 CF-PFCF 算法

5.3.1 算法设计思想

CF-PFCF 算法是以用户历史评分数据为背景，遵循协同过滤的基础流程，首先对原始用户—评分矩阵利用 PMF 技术得到近似矩阵，因为该近似矩阵与均值相比更能反映用户行为，因此作为填充数据。其次，针对性地填充用户间一方评价而另一方缺失评价的项目以充分挖掘用户特征信息、计算并集相似度。然后计算用户信任因子，分别以用户共同评分下的相似度、用户的评分次数、用户评分和被评分项目均值之差来限制填充相似度。综合用户信任因子与用户相似度，减弱相似度计算中由填充带来的假设性，加权得到最终的调和相似度。以该调和相似度由 k 近邻算法得到用户邻居集。最后利用式(5-2)进行预测评分，如图 5-2 所示为 CF-PFCF算法流程。

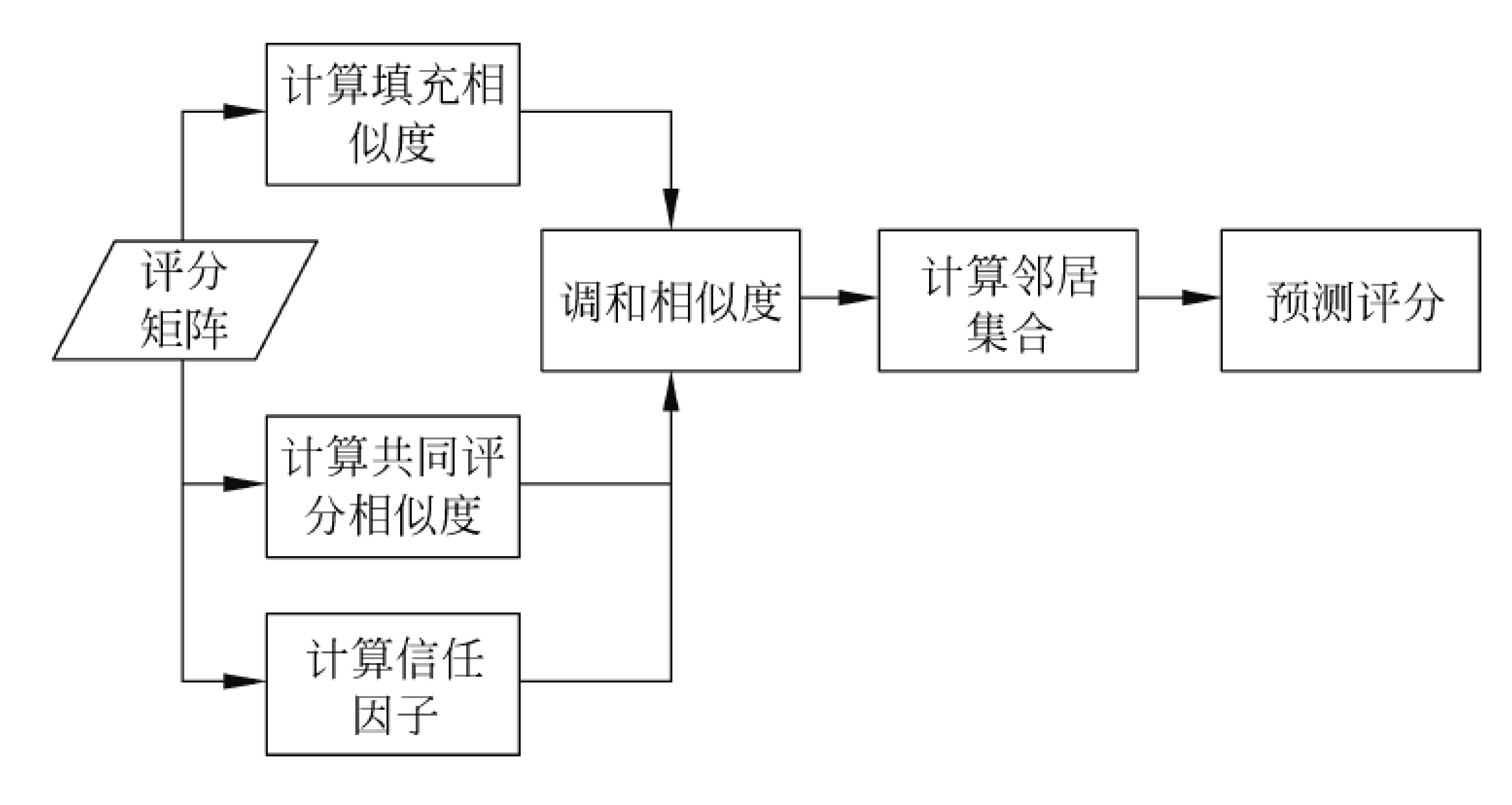


图 5-2 CF-PFCF 算法流程

1. 部分填充及相似度计算

填充稀疏矩阵的目的是更充分地利用已有评分信息计算用户之间的相似度，使得用户相似度计算更加准确。研究者已经提出许多填充评分矩阵缺失值的方法，其中最简单的填充方法以用户评分均值[16]、项目评分均值、用户评分中值、项目评分中值进行对用户并集中缺失值填充。该填充方法保证原始用户—评分矩阵的评分项参与运算，但用固定值填充导致被填充用户的特征被平均化，因此在一定

程度上减弱了被填充用户的数据特征，致使计算相似度准确率不高、甚至降低准确率。

针对以上问题提出一种部分填充算法，对于评分矩阵 R，先以 PMF 方法对其进行分解，获得近似评分矩阵$\hat{R}$，该近似评分矩阵预测用户兴趣趋势。再通过评分矩阵 R 获得用户 a 和用户 b 所评分的项目集合 V_a 和 V_b。计算集合 V_a、V_b 的并集和交集得到电影集合 $V_{a\cup b}$和 $V_{a\cap b}$。

其中集合 $V_{a\cup b}$为用户 a 或用户 b 评价过的电影，集合 $V_{a\cap b}$为用户 a 和用户 b 共同评价过的电影。由式(5-8)得到用户 a 异或用户 b 没有评价过的电影集合 $V_{a\oplus b}$。

$$V_{a\oplus b} = V_{a\cup b} - V_{a\cap b} \tag{5-8}$$

对于用户 a，依 PMF 近似评分矩阵填补其 V_a 集合中不包含 $V_{a\oplus b}$的电影评分得到补全用户评分向量 $V_{\hat{a}}$；对于用户 b，依 PMF 近似评分矩阵填补其 V_b 集合中不包含 $V_{a\oplus b}$的电影评分得到补全用户评分向量 $V_{\hat{b}}$。例如，$V_a=\{1,2,3,5\}$，$V_b=\{3,4,5,6\}$，表示用户 a 和 b 所评价过的电影需好，共同评分项仅有 $V_{a\cap b}=\{3,5\}$，计算得出集合 $V_{a\oplus b}=\{1,2,4\}$为一方评价而另一方未评价的电影，通过对$\{1,2,4\}$填补，得到填补后的评分集合 $V_{\hat{a}}=V_{\hat{b}}=\{1,2,3,4,5,6\}$，因此共同评分项由$\{3,5\}$扩展为$\{1,2,3,4,5,6\}$。

然后计算 $V_{\hat{a}}$和 $V_{\hat{b}}$的 Pearson 相似度，如式(5-9)所示。

$$\mathrm{sim}_{\mathrm{fill_Pearson}}(a,b) = \frac{\sum\limits_{v\in V_{a\cup b}} (R_{\hat{a},v} - \bar{R}_a)(R_{\hat{b},v} - \bar{R}_b)}{\sqrt{\sum\limits_{v\in V_{a\cup b}} (R_{\hat{a},v} - \bar{R}_a)^2}\sqrt{\sum\limits_{v\in V_{a\cup b}} (R_{\hat{b},v} - \bar{R}_b)^2}} \tag{5-9}$$

式中：$R_{\hat{a},v}$——补全用户评分集合 $V_{\hat{a}}$中的第 v 项电影评分值；

$R_{\hat{b},v}$——补全用户评分集合 $V_{\hat{b}}$中的第 v 项电影评分值；

$\bar{R}_a$——用户 a 在原始稀疏评分矩阵中的平均电影评分；

$\bar{R}_b$——用户 b 在原始稀疏评分矩阵中的平均电影评分。

通过该填充算法，在保证充分利用原始评分矩阵用户特征信息的前提下避免过度填充，相似度计算的假设性减弱。

2. 用户信任因子

虽然填充算法保证所有用户间相似度均可计算，但由于原始评分矩阵过于稀疏，即使进行部分填充其相似度假设性依然较强，直接用式(5-9)所计算出的相似度仍然不能反映用户之间的实际关系。因此在进行预测评分时，应考虑到多种因素对相似度的影响，这些因素称为用户信任因子。

引入基本的 Pearson 相似度对其进行加权调整，通过式(5-1)对用户间共同评价项进行计算，得出基本 Pearson 相似度 $\mathrm{sim}_{\mathrm{Pearson}}$，该相似度计算不带有任何填充项，可反映用户间真实关系。通过加权调整，可得调和后的用户相似度，如式(5-10)所示。

$$\text{sim_adj} = \alpha \text{sim}_{\text{fill_Pearson}} + (1-\alpha)\text{sim}_{\text{Pearson}} \tag{5-10}$$

用户评价的电影越多表明该用户的电影评价行为越可靠。如式(5-11)所示，用户评价等级 N_u 表示用户该特征。

$$N_u = \begin{cases} 0, & \text{num}(u) = 0 \\ \dfrac{\text{num}(u)}{\text{num}(\bar{U})} & 0 < \text{num}(u) < \text{num}(\bar{U}) \\ 1, & \text{num}(u) \geqslant \text{num}(\bar{U}) \end{cases} \tag{5-11}$$

式中：$\text{num}(u)$——用户 u 评价过的电影数目；

$\text{num}(\bar{U})$——所有用户评价过电影数目的平均值。

实际应用中一些用户喜欢评高分，一些用户喜欢评低分，甚至存在恶意评分用户，单纯用户评价等级不能衡量用户的信任度，需要加以限制。因此引入评价偏差 D_u，如式(5-12)所示。

$$D_u = \frac{|d_u|}{|Q_u|} \in [0,1] \tag{5-12}$$

式中：Q_u——用户 u 所评价过的电影集合；

d_u——用户 u 评价偏差较小的电影集合。

用户 u 对电影 i 的评价如果小于某个参考值，则认为用户 u 对电影 i 的评价偏差较小，该用户的评价无异常。通常这个参考值取电影的评价均值，通过式(5-13)进行计算。

$$|r_{u,i} - \bar{r}_i| < \varepsilon \tag{5-13}$$

如果式(5-13)成立，则 $r_{u,i} \in d_u$。通常设置 ε 为 0.5。实验显示 ε 越小，偏差要求越苛刻，取值过小会使用户丧失信任。

基于式(5-9)至式(5-12)，对相似度进行加权调整，得到综合相似度。如式(5-14)所示。

$$\text{sim}_{\text{tr}} = \alpha \text{sim}_{\text{fill_Pearson}} + (1-\alpha)\text{sim}_{\text{Pearson}} + w_1 N_u + w_2 D_u \tag{5-14}$$

关于权重值的设定，可以采用机器学习算法、专家经验等，本书采用粒子群算法，不断交叉验证，最终获取一组较优的权重值例如(0.7,0.1,0.2)。

5.3.2 CF-PFCF 算法的描述

综合以上分析，CF-PFCF 算法描述如下：

算法 5-1 CF-PFCF 算法

输入：用户—评分矩阵 R，待预测用户—评分项集合 R_{pre}，邻居数 k。

输出：用户—评分项集合 R_{pre}，其中用户 u_a 对电影 i 的评分 $\hat{r}_{a,i} \in R_{\text{pre}}$。

算法的基本流程

Step1 对稀疏的原始评分矩阵 R 进行 PMF 分解，得到 PMF 近似评分矩阵 $\hat{R}$；

Step2 遍历原始评分矩阵 R 计算相似度矩阵 $\text{sim}_{\text{fill_Pearson}}$ 和 $\text{sim}_{\text{Pearson}}$

续表

repeat

Step2.1 获得用户 u_a 和用户 u_b 各自评价电影集合的交集 $V_{a\cap b}$ 和并集 $V_{a\cup b}$；

Step2.2 依 PMF 近似评分矩阵 $\hat{R}$，填补 u_a 和 u_b 评分集合 $V_{a\cap b}$ 中缺失项，获得填补后的用户评分集合 $\hat{u}_a$ 和 $\hat{u}_b$；

Step2.3 用填补后的用户评分集合 $\hat{u}_a$、$\hat{u}_b$ 计算其相似度获得 $\mathrm{sim}_{\mathrm{fill_Pearson}}$；

until 遍历评分矩阵 R；

Step3 遍历评分矩阵 R 计算信任因子；

repeat

Step3.1 获取用户 u_a 和用户 u_b 共同评分集合，并计算共同评分下的相似度；

Step3.2 统计每位用户评价电影总数得到用户评价数目集合 $\mathrm{num}(u)$；

Step3.3 统计每位用户评价过电影序号获得用户历史评价记录集合 user_v；

Step3.4 对每部电影求其平均评价值 aver_v；

Step3.5 通过评分矩阵 R 得到对电影 i 评价过的用户集合 U_I；

until 遍历评分矩阵 R；

Step4 计算 $\mathrm{num}(u)$ 平均值得到 $\mathrm{num}(\bar{u})$。再由式(5-14)得到用户评价等级集合 N_u；

Step5 由 aver_v 和 user_v 通过式(5-12)、式(5-13)计算用户评价偏差 D_u；

Step6 利用式(5-14)计算综合相似度 $\mathrm{sim}_{\mathrm{tr}}$；

Step7 通过对 U_I 对应 $\mathrm{sim}_{\mathrm{tr}}$ 进行降序排序，取前 k 个用户作为用户 u_a 的邻居集合 U_{neighbor}；

Step8 预测用户—评分项集合 R_{pre}；

repeat

由 $\mathrm{sim}_{\mathrm{tr}}$、$U_{\mathrm{neighbor}}$ 和 R 利用式(5-2)进行评分预测，得到 $\hat{r}_{a,i}$；

until 遍历集合 R_{pre}。

算法结束

5.4　实验分析

5.4.1　数据集与误差标准

本实验采用由美国明尼苏达大学 GroupLens 创建并维护的 MovieLens 数据集。它包含 943 名用户对 1682 部电影的评分，每个用户至少评价过 20 部电影，评分集为{1,2,3,4,5}，评分越大说明用户对电影的认可度越高。用户的稀疏等级为 100 000/(943×1692)=93.7%。实验将数据集划分为两个互不相交的训练集和测试集，比例为 8∶2。

实验性能有许多评价标准，例如，查全率、均方根误差、查准率等。本书采用平均绝对误差(Mean Absolute Error，MAE)作为度量标准。假设测试集中实际评分分别为$\{p_1,p_2,p_3,\cdots,p_n\}$，算法预测的评分为$\{q_1,q_2,q_3,\cdots,q_n\}$，则 MAE 定义为式(5-15)所示。

$$\mathrm{MAE}=\frac{\sum\limits_{i=\mathrm{test}}|q_i-p_i|}{n} \tag{5-15}$$

MAE 值越小，说明算法可行性越强。

5.4.2 实验结果与性能比较

为了验证本书所使用的填充算法对传统协同过滤推荐算法的改善作用，首先将常用的几种相似度算法进行对比测试，如图 5-3 所示为余弦、调整余弦和 Pearson 相似度对比。

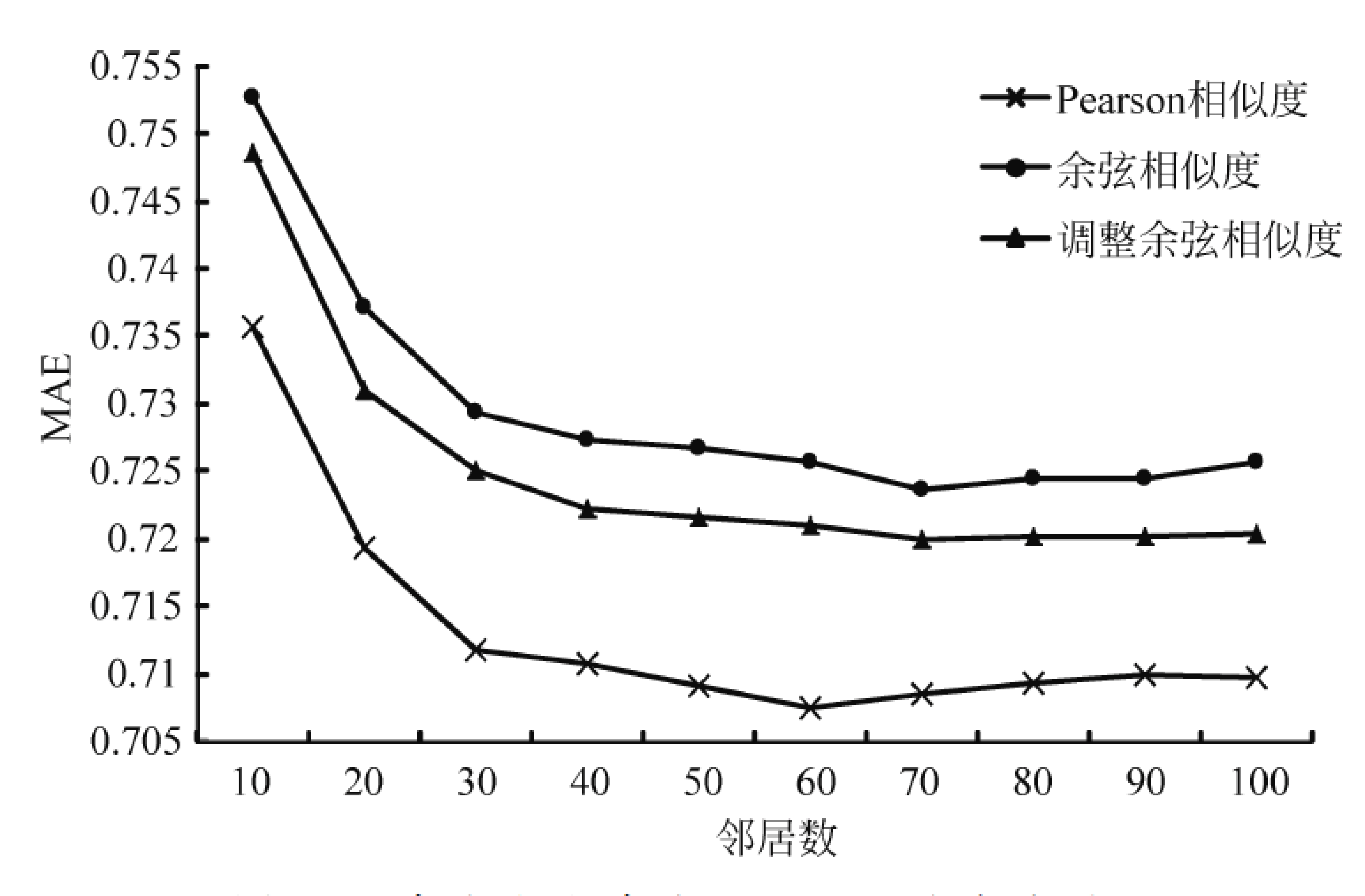


图 5-3 余弦、调整余弦和 Pearson 相似度对比

从图 5-3 可以看出，余弦相似度误差最大，Pearson 相似度误差最小，三种相似度算法随着邻居数增多，误差逐渐减小并收敛。因此本书选择以 Pearson 相似度为基础进行试验。

以 Pearson 相似度为基础，分别对评分矩阵进行 PMF 全填充、PMF 部分填充计算相似度后预测评分，如图 5-4 所示为 Pearson、全 PMF 填充和部分 PMF 填充相似度对比。其中 PMF 近似评分矩阵的 MAE=0.7505。

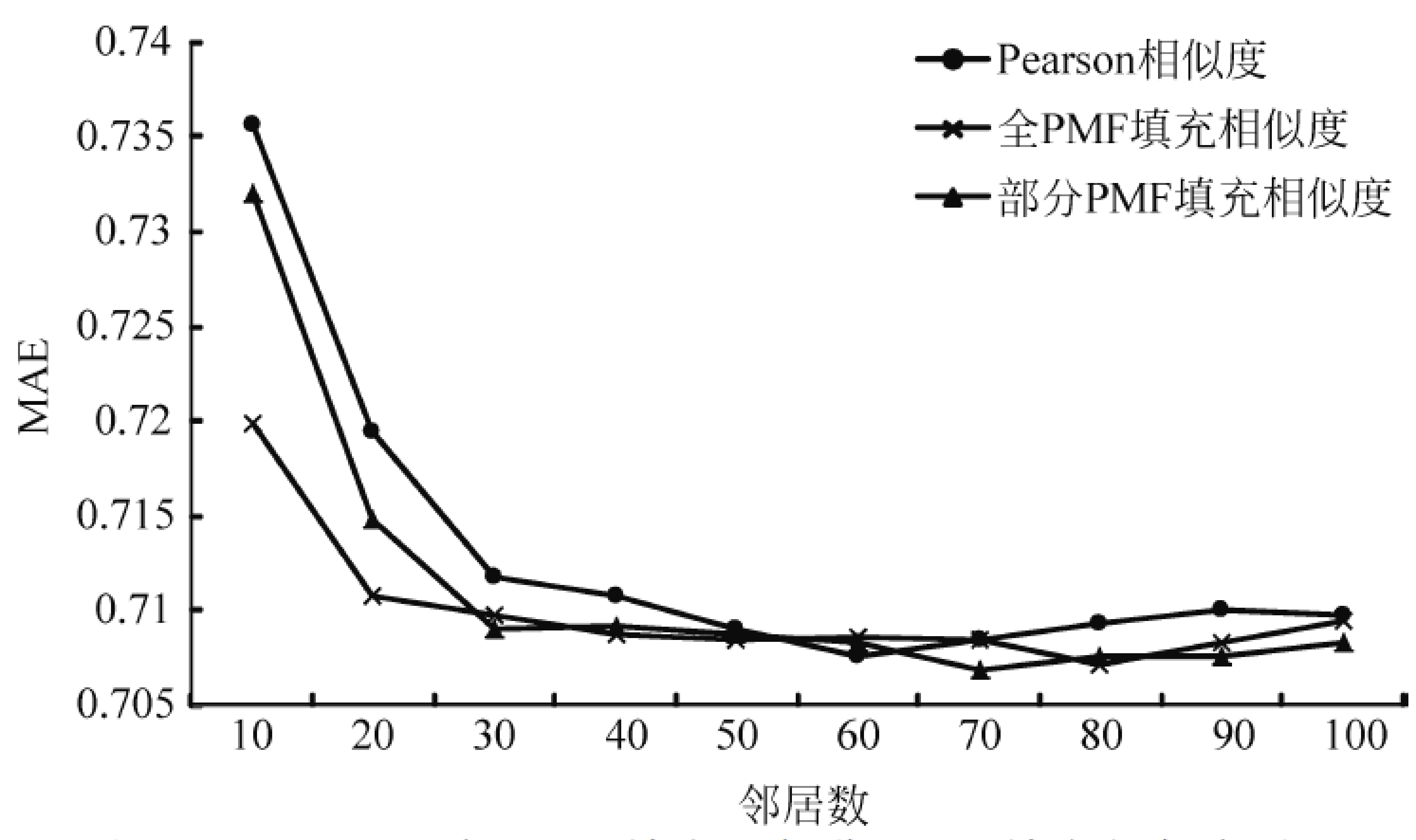


图 5-4 Pearson、全 PMF 填充和部分 PMF 填充相似度对比

从图 5-4 可以看出，基于 Pearson 相似度算法，在邻居数少的情况下填充算法精度提高，部分 PMF 填充算法在邻居数为 70 的情况下精度达到最优。图 5-3、图 5-4 表明单纯部分 PMF 填充相似度下精度提升依然不明显，为此加入用户信任因子利用式(5-16)计算用户综合相似度后预测评分。

综合相似度下预测误差与 Pearson 相似度和部分填充 PMF 相似度预测误差作对比，如图 5-5 所示为 Pearson、部分 PMF 填充和 CF-PFCF 算法对比。在信任因子限制下，预测精度显著提升并在邻居数为 40 时达到最优。

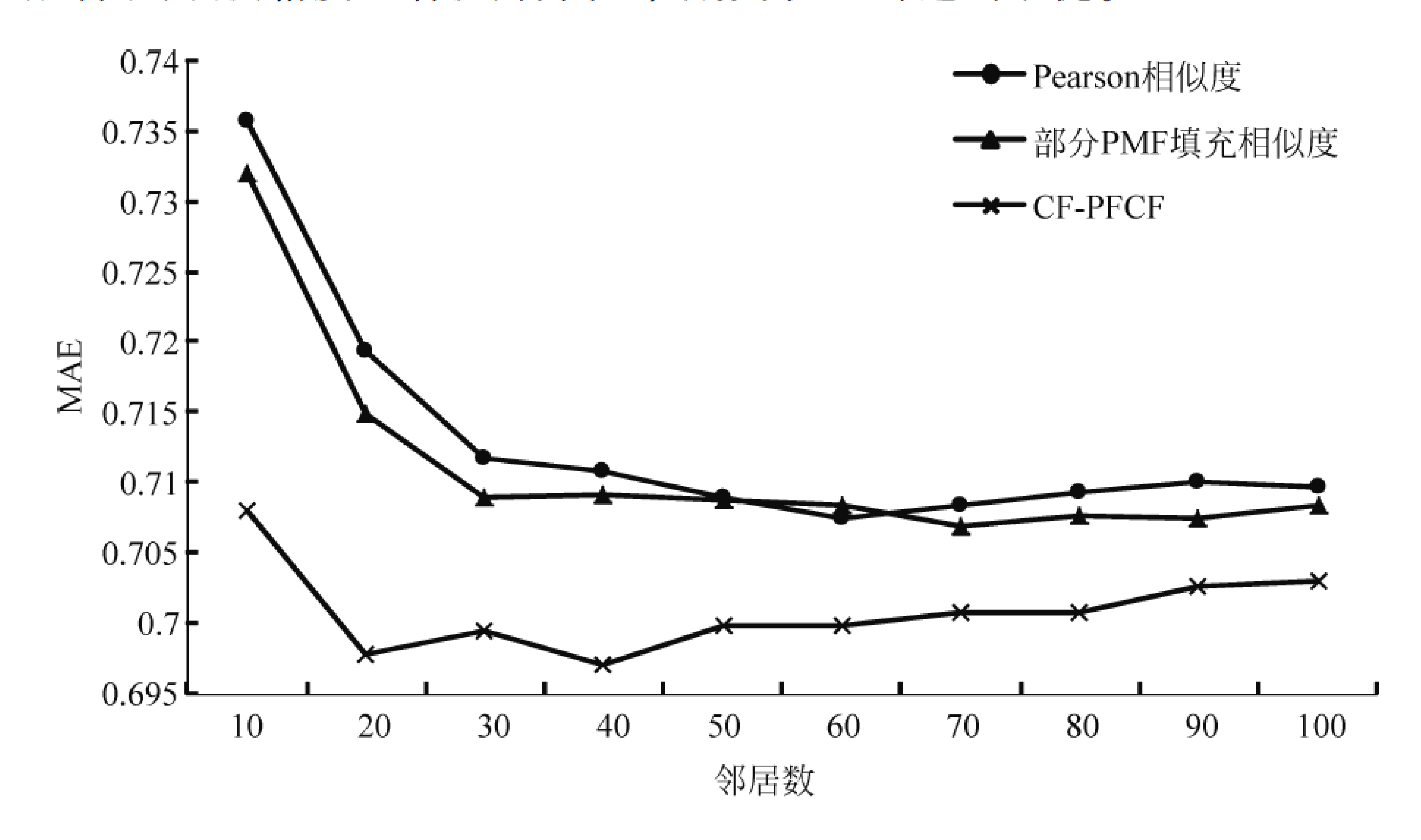


图 5-5　Pearson、部分 PMF 填充和 CF-PFCF 算法对比

把 CF-PFCF 算法与文献[13]提出的 WCF-SOFT 算法和文献[14]提出的基于信任模型填充的协同过滤的 CFTM 算法做比较，如图 5-6 所示为 CFTM、WCF-SOFT 和 CF-PFCF 算法对比。

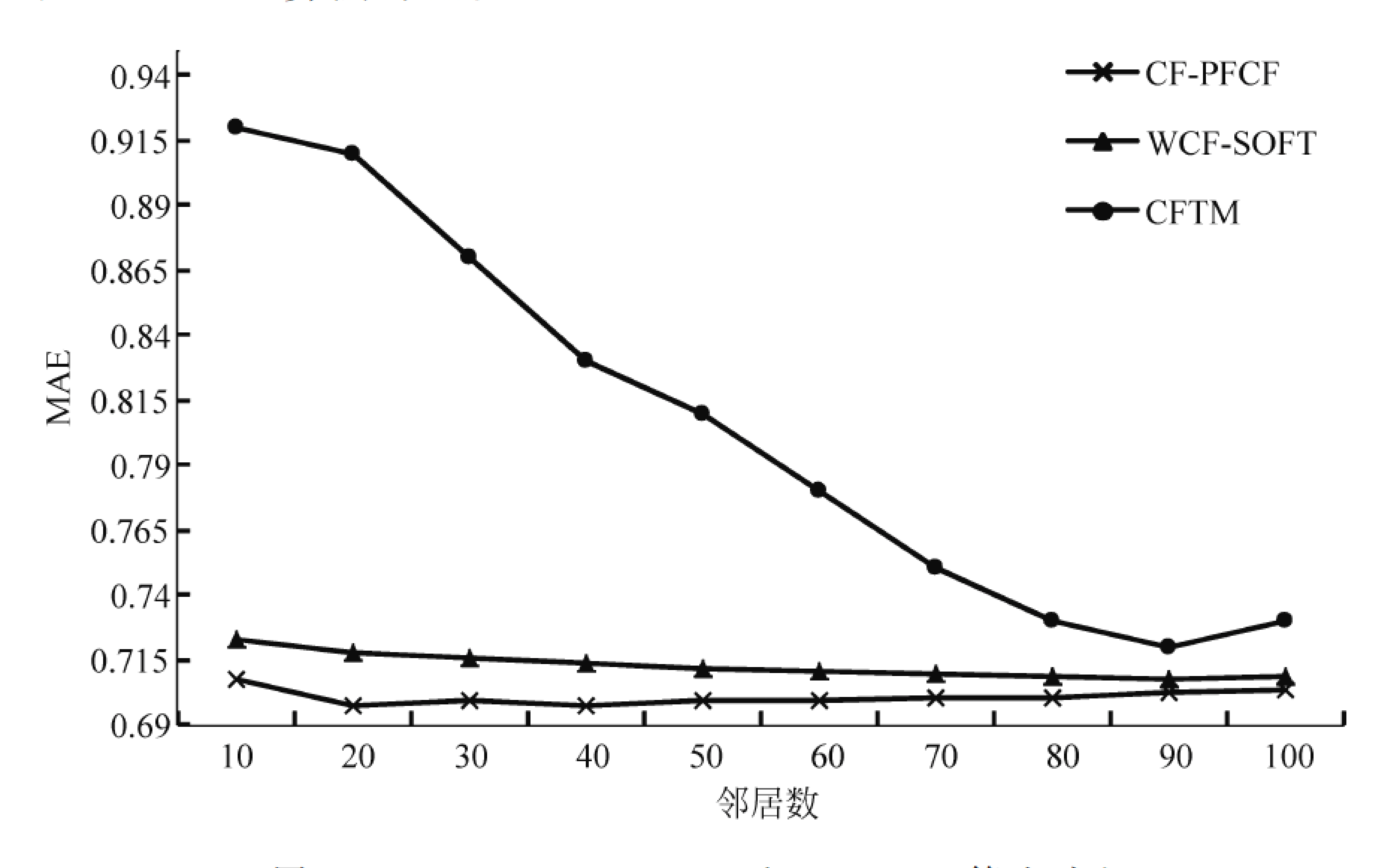


图 5-6　CFTM、WCF-SOFT 和 CF-PFCF 算法对比

从图 5-6 可以看出,本书算法精度明显优于 WCF-SOFT、CFTM 算法。由于 CFTM 算法主要计算信任因子没有对相似度进行恰当改进,所以 CFTM 算法误差较大,在邻居数为 90 时达到最优,邻居数为 100 时误差上升。WCF-SOFT 算法稳定性较强,由于算法运用了填充技术并加以信任因子限制,所以预测误差明显下降。本书算法对 WCF-SOFT 算法的填充部分进行改进,以部分填充代替全局填充降低填充评分的假设性成分,并加以共同评分相似度和信任因子限制。

本章小结

本章对填充矩阵和信任因子进行了研究,在高维稀疏的数据和基于用户的协同过滤推荐算法的基础上提出部分相似度填充和信任因子限制。部分填充保证用户特征充分利用的前提下避免过度填充,解决了高维稀疏评分矩阵用户间共同评分稀少甚至缺失的问题,并且对填充算法的假设性进行限制。尽管算法提高了整体精确度,但由于用户信任因子的影响,该算法随着邻居数增多精确度非单调下降,下一步需要研究如何增强算法稳定性,使误差随着邻居数增加单调递减并收敛。

参考文献

[1] Adomavicius G, Tuzhilin A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions[J]. IEEE Transactions on Knowledge & Data Engineering, 2005, 17(6): 734-749.

[2] Wang J, Yin J. Enhancing Accuracy of User-based Collaborative Filtering Recommendation Algorithm in Social Network[C]//International Conference on System Science, Engineering Design and Manufacturing Informatization, 2012: 6082-6086.

[3] Licia Capra, Neal Lathia, Stephen Hailes. Trust-Based Collaborative Filtering[J]. Trust Management Ⅱ, 2008.

[4] Sarwar B, Karypis G, Konstan J, et al. Item-based Collaborative Filtering Recommendation Algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 285-295.

[5] Salakhutdinov B R, Mnih A. Probabilistic Matrix Factorization[C]//International Conference on Machine Learning, 2012: 880-887.

[6] Lee C, Lin C. Large-Scale Linear RankSVM[J]. Neural Computation, 2014, 26(4): 781-817.

[7] Lin H, Yang X, Wang W, et al. A Performance Weighted Collaborative Filtering Algorithm for Personalized Radiology Education[J]. Journal of Biomedical Informatics, 2014, 51: 107-113.

[8] Bokde D, Girase S, Mukhopadhyay D. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey[J]. Procedia Computer Science, 2015, 49(1): 136-146.

[9] Zhu Qi-wei, Chen Jia-qi. An Improved Collaborative Filtering Recommendation Algorithm

Based on an Improved Similarity Calculation Method[J]. Information Technology, 2015(3): 13-16(in Chinese).

[10] Fan Bo, Cheng Jiu-jun. Collaborative Filtering Recommendation Algorithm Based on User's Multi-similarity[J]. Computer Science, 2012, 39(1): 23-26(in Chinese).

[11] Bobadilla J, Serradilla F, Bernal J. A New Collaborative Filtering Metric that Improves the Behavior of Recommender Systems[J]. Knowledge-Based Systems, 2010, 23(6): 520-528.

[12] Sun Xiao-hua, Chen Hong, Kong Fan-sheng. Combining Singular Value Decomposition and Neighbor-based Method in Collaborative Filtering[J]. Application Research of Computers, 2006, 23(9): 206-208(in Chinese).

[13] Hao Li-yan, Wang Jing. Collaborative Filtering Recommendation Algorithm Based on Filling and Similarity Confidence Factor[J]. Journal of Computer Applications, 2013, 33(3): 834-837(in Chinese).

[14] Yang Xing-yao, Yu Jiong, Turgun Ibrahim, et al. Collaborative Filtering Recommendation Model Based on Trust Model Filling[J]. Computer Engineering, 2015(5): 6-13(in Chinese).

[15] Yang W F, Wang M, Chen Z. Fast Probabilistic Matrix Factorization for recommender system[C]//IEEE International Conference on Mechatronics and Automation, 2014: 1889-1894.

[16] Mazumder R, Hastie T, Tibshirani R. Spectral Regularization Algorithms for Learning Large Incomplete Matrices[J]. Journal of Machine Learning Research, 2010, 11(11): 2287-2322.

[17] Wang Q M, Liu X, Zhu R, et al. A New Personalized Recommendation Algorithm of Combining Content-based and Collaborative Filters[J]. Computer & Modernization, 2013, 1(8): 64-67.

[18] Tsai C F, Hung C. Cluster Ensembles in Collaborative Recommendation[J]. Applied Soft Computing, 2012, 12(4): 1417-1425.

[19] 范敏敏. 非负矩阵分解与聚类方法在个性化推荐系统中的应用研究[D]. 南昌：华东交通大学，2012.

[20] 居斌，钱沄涛，叶敏超. 基于结构投影非负矩阵分解的协同过滤算法[J]. 浙江大学学报工学版，2015，49(7)：1319-1325.

[21] 张月蓉. 基于混合推荐的电影推荐系统的研究与实现[D]. 合肥：安徽大学，2013.

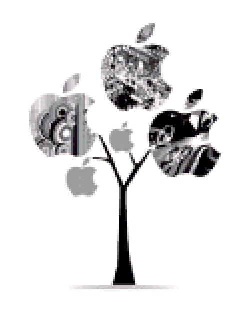
[22] 孙海峰，甘明鑫，刘鑫，等. 国外电影推荐系统网站研究与评述[J]. 计算机应用，2013，A02(z2)：119-124.

[23] 王静. 基于隐式社会网络的电影推荐系统研究[D]. 北京：北京交通大学，2014.

[24] 李刚. 整合 Struts + Hibernate + Spring 应用开发详解[M]. 北京：清华大学出版社，2007.

[25] 李绍平，彭志平. S2SH：一种 Web 应用框架及其实现[J]. 计算机技术与发展，2009，08(8)：117-119.

第 6 章 基于偏置信息的改进概率矩阵分解算法研究



针对个性化推荐过程中高维稀疏性问题，本章提出一种奇异值分解和带偏置概率矩阵分解相结合的推荐方法。首先获取用户—项目评分矩阵，通过奇异值分解来初始化用户和项目的潜在因子矩阵，然后将偏置信息融入到概率矩阵分解算法中以提升推荐的精度，最后结合最大似然估计把评分预测问题转化为最优化问题，通过随机梯度下降来求解最优化问题。在三个公开数据集的实验结果表明，本章提出的算法能够有效地提升推荐精度，缓解由数据高维稀疏性带来的推荐精度不高的问题。

6.1 引言

个性化推荐算法的任务是利用用户—项目评分矩阵中的已知元素来预测未知元素的评分值并将预测评分高的项推荐给用户，但是在研究过程中面临诸多的问题，其中最为典型的是由数据的高维稀疏性带来的推荐精度不高的问题。

在推荐算法的研究中，用户和项目的数量通常都是数万甚至数百万的，如果仍然用适用于低维数据的相似度度量方式来处理这些高维的特征数据，将得不到理想的结果，即所谓的“维度灾难”。同时，用户和项目的数量是十分巨大的，获取的用户—项目评分矩阵每一行代表一个用户，每一列代表一个物品，只是这个矩阵可能是非常稀疏的，即已知的评分只占总评分数量的很少一部分，一般采用稀疏度来度量数据集的稀疏性，其定义为用户评分数据矩阵中未评分条目所占的百分比。例如 Epinions 数据集共包含 22 166 个用户和 29 277 个项目，评分记录只有 922 267 条，若采用 Pearson 相似度等传统度量方法，任意两个用户的相似度都近似为 0，也就是说根本区分不开，数据集的稀疏度为 22 166×296 277/922 267≈0.000 14。

针对数据的高维稀疏性带来的问题，许多研究者提出可以利用填充算法、基于聚类的算法来解决，例如文献[1]提出了基于项目的 SlopeOne 数据填充的预测算法，该算法根据项目之间的评分差异来预测未知评分；文献[2]提出一种基于用户模糊聚类的协同过滤推荐算法，该算法首先利用用户情景信息来对用户聚类，然后

在不同的聚类簇中应用填充算法来预测评分。

然而上述算法没有考虑到用户的兴趣差异，不能体现用户间的爱好区别，尤其对于高维稀疏数据推荐效果往往难以得到保证，因此本章提出一种融合偏置信息和奇异值分解（Singular Value Decomposition，SVD）的概率矩阵分解（Probabilistic Matrix Factorization，PMF）算法，称为 BSPMF，利用分解后的两个低维矩阵对原矩阵中的未知评分进行预测，在降维的同时能够缓解由数据的高维稀疏性带来的推荐精度不高的问题。

6.2 相关工作

6.2.1 矩阵分解模型

矩阵分解模型假设用户对项目的评分受到若干潜在因子的影响，将用户和项目映射到一个共同的潜在因子空间，这其实与实际情况是相符的，人们在购物时往往也是受到很多因素的影响，如价格、品牌、款式等。不过在该类模型中这些潜在因子是很难解释出其究竟是哪个具体的因素，也正因为此原因矩阵分解模型又可被称作隐语义模型。

基于矩阵分解的算法将原始的高维度的评分矩阵 R_{mn} 近似分解成两个低维度的矩阵的乘积，如式(6-1)所示。

$$R_{mn} \approx U_{km}^{\mathrm{T}} V_{kn} \tag{6-1}$$

式中：m——用户的个数；

n——项目的个数；

$k \ll \min(m,n)$——潜在因子的数量；

U_{km}^{T} 和 V_{kn}——两个低维矩的用户潜在因子矩阵和项目潜在因子矩阵。

可通过迭代训练来使得 U_{km}^{T} 和 V_{kn} 的内积不断地逼近原始评分矩阵，同时得到 U_{km}^{T} 和 V_{kn} 后还可以对用户没有评分的项目进行评分预测。

基于矩阵分解的算法是一种学习型算法，通过优化预先设定的目标函数而得到全局最优解，而且由于潜在因子的数量 $k \ll \min(m,n)$，算法的离线计算的空间复杂度低，这在当今大数据的环境下具有很强的实用价值；同时，由于该算法有一个全局的目标函数，使得算法的预测准确率高。

1. 奇异值分解

奇异值分解是一种常用的降低数据维度的矩阵分解算法，是由 Scott Deerwester 等人在 1990 年提出。SVD 在推荐算法研究中有着广泛的应用。

如式(6-2)所示，SVD 将原始矩阵 R 近似分解为 3 个矩阵的乘积。

$$R_{mn} \approx U_{mm} S_{mn} V_{mm}^{\mathrm{T}} \tag{6-2}$$

式中：U——左奇异矩阵；

V——右奇异矩阵；

S——奇异值矩阵。

算法 6-1　基于 SVD 的推荐算法

输入：用户—项目评分矩阵 R_{mn} 和潜在因子维度向量 $kVector$，最大迭代次数 maxIter，前后两次迭代的均方根误差阈值 RMSEThreshold$=10^{-6}$。

输出：用户潜在因子矩阵 U_{mk}，前 k 大奇异值矩阵 S_{kk}、项目潜在因子矩阵 V_{nk} 以及最优潜在因子的维度 k。

算法的基本流程如下：

步骤 1：将用户—项目评分矩阵 R_{mn} 中的缺失值用对应项目的均值填充；

步骤 2：for $kVector$ 中的每一个样本 $kMay$ do

```
            for 重复迭代直到 maxIter do
                [U_mm, S_mn, V_nn]=SVD(R_mn, k);
                选择 S_mn 中前 kMay 大奇异值得到 √S(k);
                        U_mk = U_mm * √S(k);
                        V_nk = √S(k) * V_nn^T;
                真实评分数据索引 R_L=R_mn>0;
均方根误差 RMSE=norm((U_mk V_nk^T − R_mn)×R_L,'fro')/sqrt(sum(R_L));
                if 上一次和下一次迭代的 RMSE 大于 RMSEThreshold
                    continue;
                else
                    k=kMay;
                End if
            End for
            if 本次潜在因子维度下的 RMSE 大于上一次
                更新 k;
            End if
        End for
```

步骤 3：End

算法结束

---

2. 概率矩阵分解模型

概率矩阵分解模型是矩阵分解模型中应用非常成功的一个推荐模型，是由 H. Shan 和 A. Banerjee 于 2007 年提出，该模型在 Netflix 推荐竞赛以及在的 KDD Cup 竞赛中都取得了非常优异的成绩。如图 6-1 所示为 PMF 的概率图模型。

概率矩阵分解的基本思想是在矩阵分解的基础上引入概率的思想，假设用户和商品的特征向量矩阵都符合高斯分布，见式(6-3)。

$$p(U \mid \sigma_U^2) = \prod_{i=1}^{N} N(U_i \mid 0,\sigma_U^2 I),\quad p(V \mid \sigma_V^2) = \prod_{j=1}^{M} N(V_j \mid 0,\sigma_V^2 I) \quad (6\text{-}3)$$

基于这个假设，用户对商品的喜好程度就是一系列概率的组合问题，见式(6-4)。

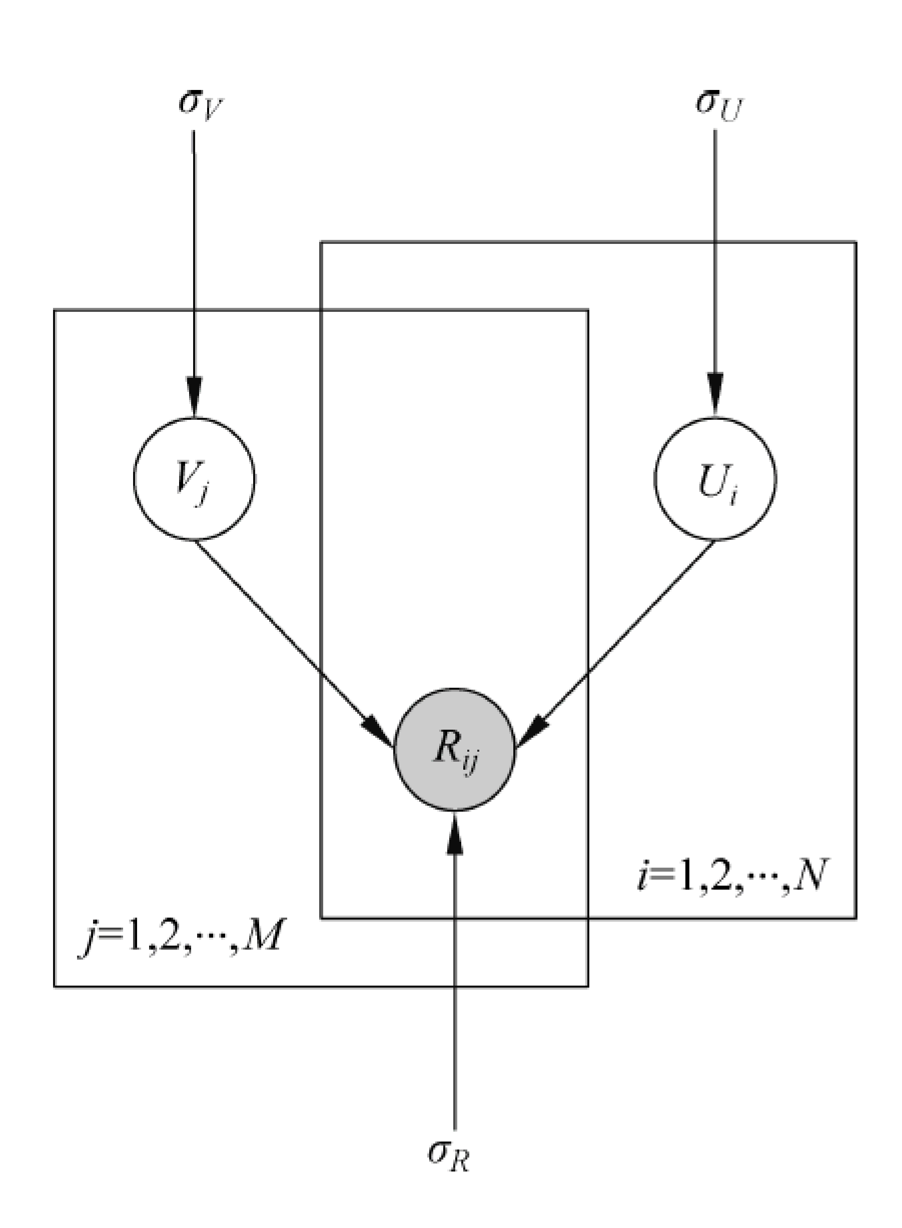


图 6-1　PMF 概率图模型

$$p(R \mid U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[N(R_{ij} \mid U_i^{\mathrm{T}}V_j,\sigma^2)]^{I_{ij}} \tag{6-4}$$

式中：$p(A|B)$——事件 B 发生的情况下事件 A 发生的条件概率；

$N(x|\mu,\sigma^2)$——期望为 μ，方差为 σ 的高斯分布。

I 为指示函数，如果用户 i 选择了商品 j，$I_{ij}=1$，否则为 0。

利用贝叶斯推导，可得用户和物品的隐式特征的后验概率，如式(6-5)所示。

$$p(U,V \mid R,\sigma^2,\sigma_V^2,\sigma_U^2) \propto p(R \mid U,V,\sigma^2)\times p(U \mid \sigma_U^2)\times p(V \mid \sigma_V^2) \tag{6-5}$$

对上述预测公式取对数，可以得到式(6-6)。

$$\begin{aligned}\ln p(U,V \mid R,\sigma^2,\sigma_V^2,\sigma_U^2) = &-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}(R_{ij}-U_i^{\mathrm{T}}V_j)^2-\frac{1}{2\sigma_U^2}\sum_{i=1}^{N}U_i^{\mathrm{T}}U_i-\\&\frac{1}{2\sigma_V^2}\sum_{j=1}^{M}V_j^{\mathrm{T}}V_j-\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\right)\ln\sigma^2+\right.\\&\left.ND\ln\sigma_U^2+MD\ln\sigma_V^2\right)+C\end{aligned} \tag{6-6}$$

式中：C——一个不依赖于参数的常数。

最大化 U 和 V 的后验概率等于最小化公式(6-7)。

$$\begin{aligned}\arg\min_{U,V} = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}(R_{ij}-U_i^{\mathrm{T}}V_i)^2+\\&\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{\mathrm{Fro}}^2+\frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|_{\mathrm{Fro}}^2\end{aligned} \tag{6-7}$$

式中：Fro——矩阵的 F 范数。

然后利用 SGD 求解式(6-7)。

6.2.2 Baseline 预测

观测到的评分数据有一些是和用户或物品无关的因素产生的效果，即一部分因素是和用户对物品的喜好无关而只取决于用户或物品本身特性。例如，乐观型用户的评分行为普遍偏高，而批判型用户的评分记录普遍偏低，也就是说即使这两类用户对同一项目的评分相同，但是对该物品的喜好程度却并不一样。同理，对同一件(类、种)物品来说，以电影为例，受大众欢迎的电影得到的评分普遍偏高，而一些烂片的评分普遍偏低，这些因素都是独立于用户或产品的因素，而和用户对产品的的喜好无关。本章将这些独立于用户或独立于物品的因素称为偏置(Bias)信息[17]，加入偏置信息的评分预测算法如式(6-8)所示。

$$\hat{R}_{ij} = \mu + bu(i) + bvi(j) \tag{6-8}$$

式中：$\hat{R}_{ij}$——用户 i 对项目 j 的预测评分；

μ——数据集的总体偏置信息；

$bu(i)$——用户 i 的偏置信息；

$bi(j)$——项目 j 的偏置信息。

例如，要预测用户 user1 对电影 movie1 的打分，假设电影数据集总体偏置 μ 为 3.4 分，电影 movie1 的口碑比其他电影高 0.9 分，即 bi(movie1)=0.9；另一方面，用户 user1 是一个乐观的用户，一般偏向于给电影打低分(0.3 分)，即 bu(user1)=−0.3，那么用户 user1 对电影 movie1 的预测打分为 3.4+0.9−0.3=4 分。

加入偏置信息后，目标函数如式(6-9)。

$$\underset{bu,bi}{\arg\min} E(bu,bi) = \sum_{u,i}(R_{ui} - \mu - bu(u) - bi(i))^2 + \lambda_1 \sum_u bu(u)^2 + \lambda_2 \sum_i bi(i)^2 \tag{6-9}$$

在实际应用中往往根据经验似然采用式(6-10)的方法来求解 bi 和 bu 的值。

$$\begin{cases} bi = \dfrac{\sum\limits_{u \in R(i)} (R_{ui} - \mu)}{\lambda_1 + |R(i)|} \\ bu = \dfrac{\sum\limits_{i \in R(u)} (R_{ui} - \mu - bi)}{\lambda_2 + |R(u)|} \end{cases} \tag{6-10}$$

式中：u——某一用户；

i——某一项目；

$R(i)$——评价过项目 i 的用户集合；

$R(u)$——用户 u 评价过的项目集合；

λ_1 和 λ_2——压缩系数，需要实验确定。

6.3　算法流程

首先用算法 6-1 对实验数据集进行训练，然后将训练得到的 U 和 V 作为带偏置概率矩阵分解的初始值。同时由于 Bias 的存在，提出一个近似的带偏置的预测评分算法，如式(6-11)所示。

$$\hat{R}_{ij} = \mu + bu(i) + bi(j) + U_i^{\mathrm{T}} V_j \tag{6-11}$$

式中：$\hat{R}_{ij}$——根据本章的 BSPMF 算法获得的预测评分；

μ——数据集的总体偏置；

bu——各个用户的偏置向量；

bi——各个项目的偏置向量。

目标函数采用式(6-12)。

$$\begin{aligned}\underset{U,V,bu,bi}{\operatorname{argmin}} E(U,V,bu,bi) = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\,(R_{ij} - \mu - bu(i) - bi(j) - U_i^{\mathrm{T}} V_j)^2 + \\ &\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{\mathrm{Fro}}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|_{\mathrm{Fro}}^2 + \\ &\frac{\lambda_{bu}}{2}\sum_{i=1}^{N}\|bu(i)\|^2 + \frac{\lambda_{bi}}{2}\sum_{i=1}^{N}\|bi(j)\|^2\end{aligned} \tag{6-12}$$

算法 6-2　基于 BSPMF 的推荐算法

输入：用户—项目评分矩阵 R_{mn}，用户项目的潜在因子矩阵 U_{mk} 和 V_{nk}，最优潜在因子维度 k，算法最大迭代次数 maxEpoch，前后均方根误差阈值 Threshold。

输出：用户潜在因子矩阵 U 和项目潜在因子矩阵 V，用户偏置 bu 和项目偏置 bi。

算法的基本流程如下：

步骤 1：根据算法 6-1 训练得到 U_{mk} 和 V_{nk} 和最优潜在因子维度 k，$U=U_{mk}^{\mathrm{T}}$，$V=V_{nk}$；

步骤 2：评分值 sum=0，评分记录条数 num=0；

步骤 3：根据 R_{mn} 计算得到数据集的总体偏置 μ

　　for 对于 R_{mn} 中的每一条评分记录 do

　　　sum=sum+R_{mn}(num)；

　　　num=num+1；

　　End for

　　μ=sum/num；

步骤 4：初始化用户偏置向量 bu=randn(m,k)和项目偏置向量 bi=randn(n,k)；

步骤 5：计算式(6-12)的偏导，如式(6-13)所示：

续表

$$
\begin{cases}
\frac{\partial E}{\partial U} = \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - \mu - bu(i) - bi(j) - U_i^{\mathrm{T}}V_j) * V_j + \lambda_U \sum_{i=1}^{N} \| U_i \| \\
\frac{\partial E}{\partial V} = \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - \mu - bu(i) - bi(j) - U_i^{\mathrm{T}}V_j) * U_i^{\mathrm{T}} + \lambda_V \sum_{j=1}^{M} \| V_j \| \\
\frac{\partial E}{\partial bu} = \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - \mu - bu(i) - bi(j) - U_i^{\mathrm{T}}V_j) + \lambda_{bu} \sum_{i=1}^{N} \| bu(i) \| \\
\frac{\partial E}{\partial bi} = \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - \mu - bu(i) - bi(j) - U_i^{\mathrm{T}}V_j) + \lambda_{bi} \sum_{j=1}^{M} \| bi(j) \|
\end{cases}
\tag{6-13}
$$

结合 SGD 对式(6-12)进行训练；

for 循环迭代直到最大迭代次数 maxEpoch do

随机选择一条评分记录，按式(6-14)进行更新：

$$
\begin{cases}
U = U - \alpha * \frac{\partial E}{\partial U} \\
V = V - \alpha * \frac{\partial E}{\partial V} \\
bu = bu - \alpha * \frac{\partial E}{\partial bu} \\
bi = bi - \alpha * \frac{\partial E}{\partial bi}
\end{cases}
\tag{6-14}
$$

式中：α——学习因子，需要实验确定，一般选择 0.001；

真实评分数据索引 $R_L = R_{mn} > 0$；

计算均方根误差 RMSE=norm(($U^{\mathrm{T}}V - R_{mn}$) * R_L, 'fro')/sqrt(sum(R_L))；

if 上一次和下一次迭代的 RMSE 大于 Threshold

continue；

else

算法结束；

End if

End for

算法结束

6.4 实验分析

本章通过实验来检验提出算法的推荐质量，并讨论关于 Baseline 预测算法的参数、潜在因子的维度对推荐结果的影响，同时将本章提出的基于 BSPMF 模型的推荐算法与经典的 userMean 算法、Baseline 预测算法、PMF 算法、BiasPMF 算法做比较。

6.4.1 实验所用数据集

本次实验分别在 Epinions、Ciao、Movielens 三个数据集进行，这三个数据集都包含了用户对项目的评分信息，评分值为 1～5 的离散值，数据集的具体信息如表 6-1 所示。

表 6-1 实验所用数据集信息

数 据 集 名	用 户 数 目	项 目 数 目	评分记录数
Movielens	943	1682	100 000
Ciao	7375	106 797	284 086
Epinions	22 166	296 277	922 267

6.4.2 实验环境配置

本章实验中主要采用 MATLAB 语言编写所提出的方法，具体的实验配置包括硬件环境配置和软件环境配置两部分。

（1）硬件环境配置：Intel 酷睿 i3-4150 处理器，主频 305GHz，4G 内存，500G 硬盘。

（2）软件环境配置：开发工具 MATLAB R2014a，操作系统为 Windows7 旗舰版。

6.4.3 实验评价标准

为检验本章提出算法的推荐质量，实验采用均方根误差（ Root Mean Square Error，RMSE）作为度量标准[式(6-15)]，也是目前最常用的一种推荐质量度量方法，通过计算预测的用户评分与实际的用户评分之间的误差平方和的均方根，来表示预测的准确性，RMSE 值越小，推荐质量就越好。

$$\text{RMSE} = \sqrt{\frac{\sum_{u,i}(\hat{R}_{ui} - R_{ui})^2}{N}} \tag{6-15}$$

式中：$\hat{R}_{ui}$——预测的的评分；

R_{ui}——测试集中的实际评分；

N——测试集包含的数据条数。

6.4.4 实验结果及分析

进行 BaseLine 预测时，首先，将数据集的 90%划分为训练集，其余作为测试集。其次，做 Baseline 预测。选择 Baseline 做预测的目的在于该算法训练时间短，预测精度较高，对于不同的数据集，一般可通过实验训练找到参数的最优设定。

如图 6-2 所示为 MovieLens 数据集的 Baseline 预测，其中 $\lambda_1=2$，$\lambda_2=5$，RMSE 最小为 0.9559。

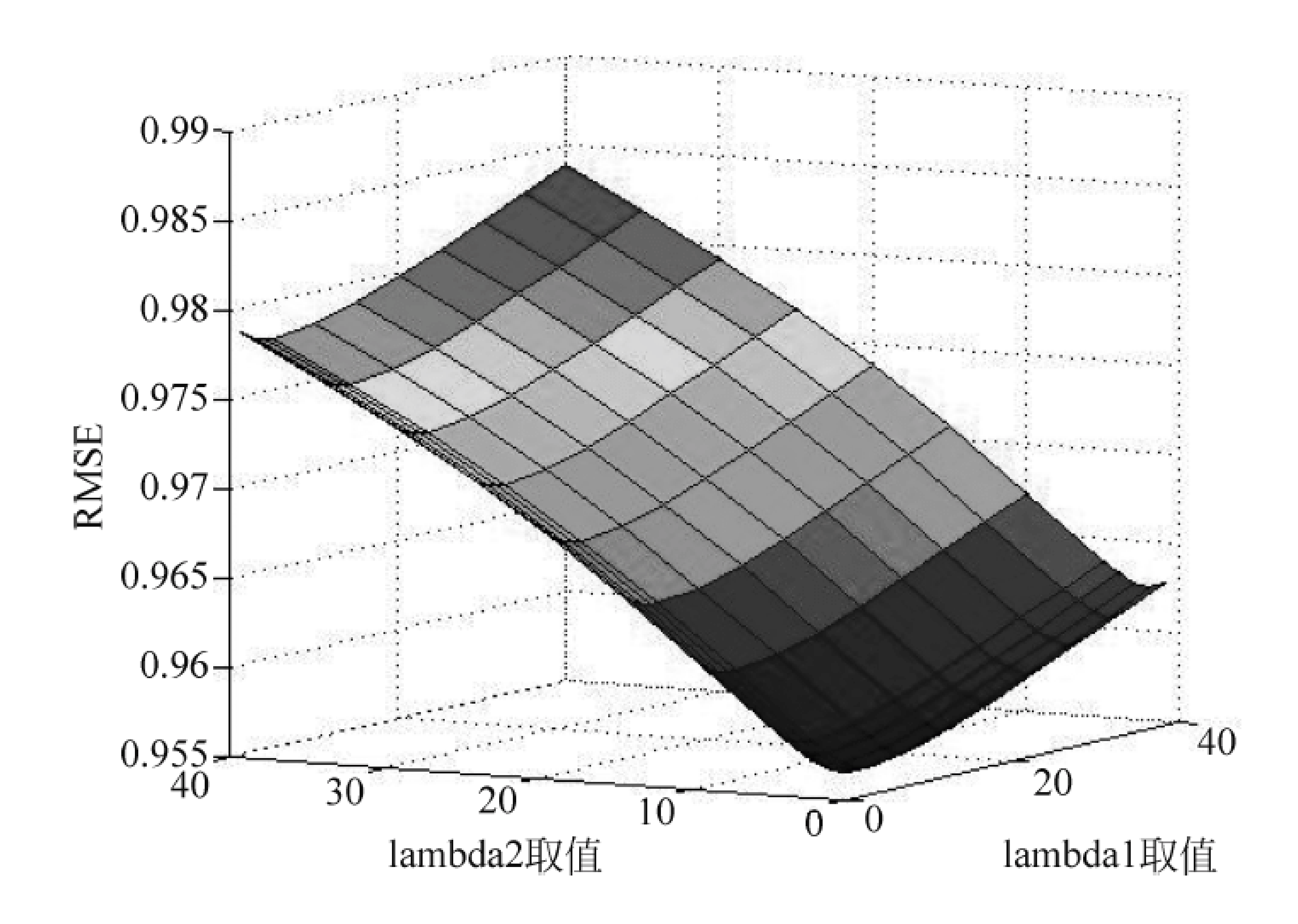


图 6-2 MovieLens 数据集的 Baseline 预测

如图 6-3 所示为 Ciao 数据集上做的 Baseline 预测，其中 $\lambda_1=60$，$\lambda_2=40$，RMSE 最小为 1.0279。

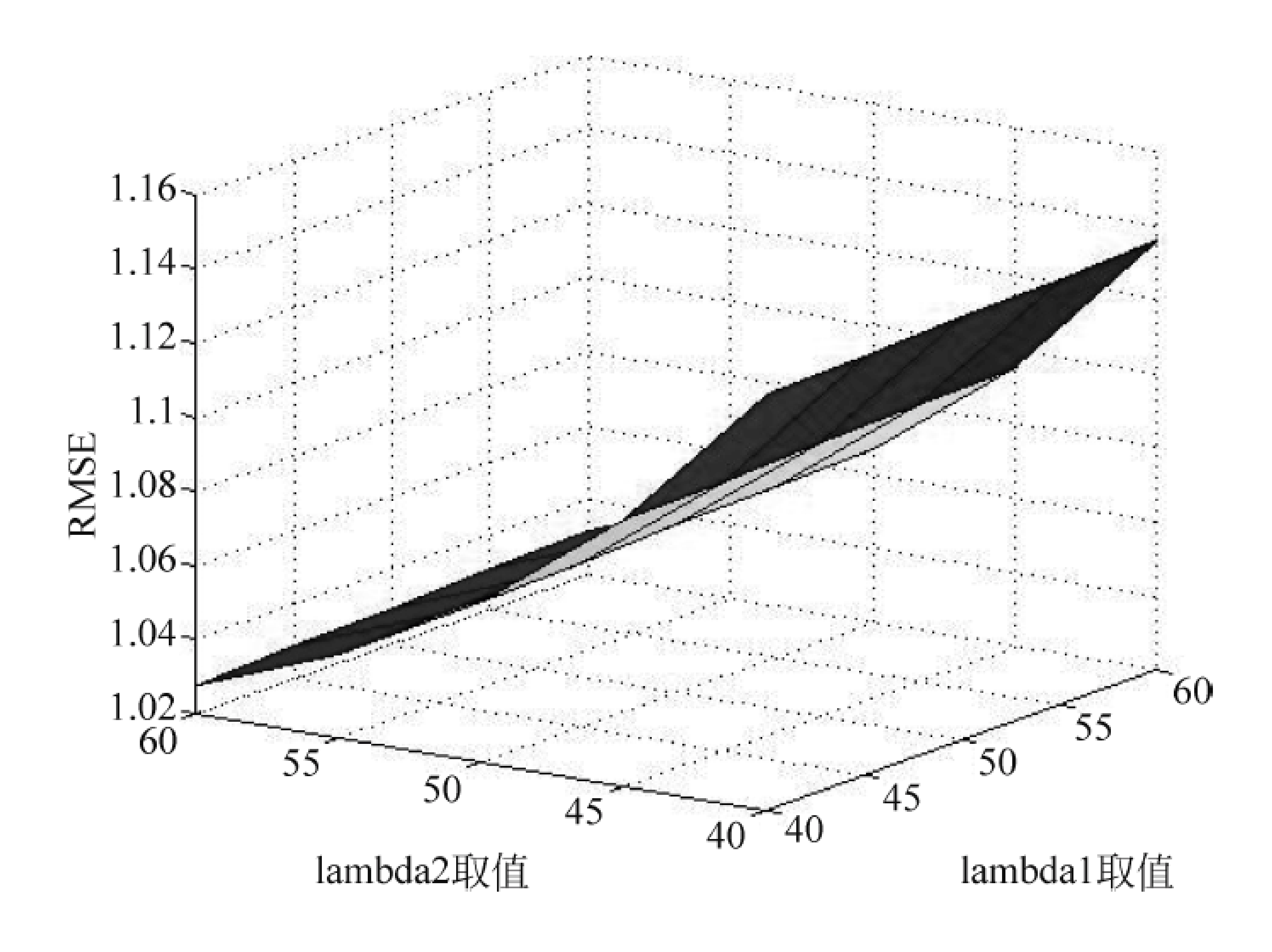


图 6-3 Ciao 数据集的 Baseline 预测

如图 6-4 所示为 Epinions 数据集上的 Baseline 预测，其中 $\lambda_1=55$，$\lambda_2=45$，RMSE 最小为 1.1038。

然后，在已知 RMSE 最小值的情况下，确定潜在因子的维度 k，如图 6-5 至图 6-7 所示分别为 MovieLens 数据集潜在因子维度、Ciao 数据集潜在因子维度和 Epinions 数据集潜在因子维度。

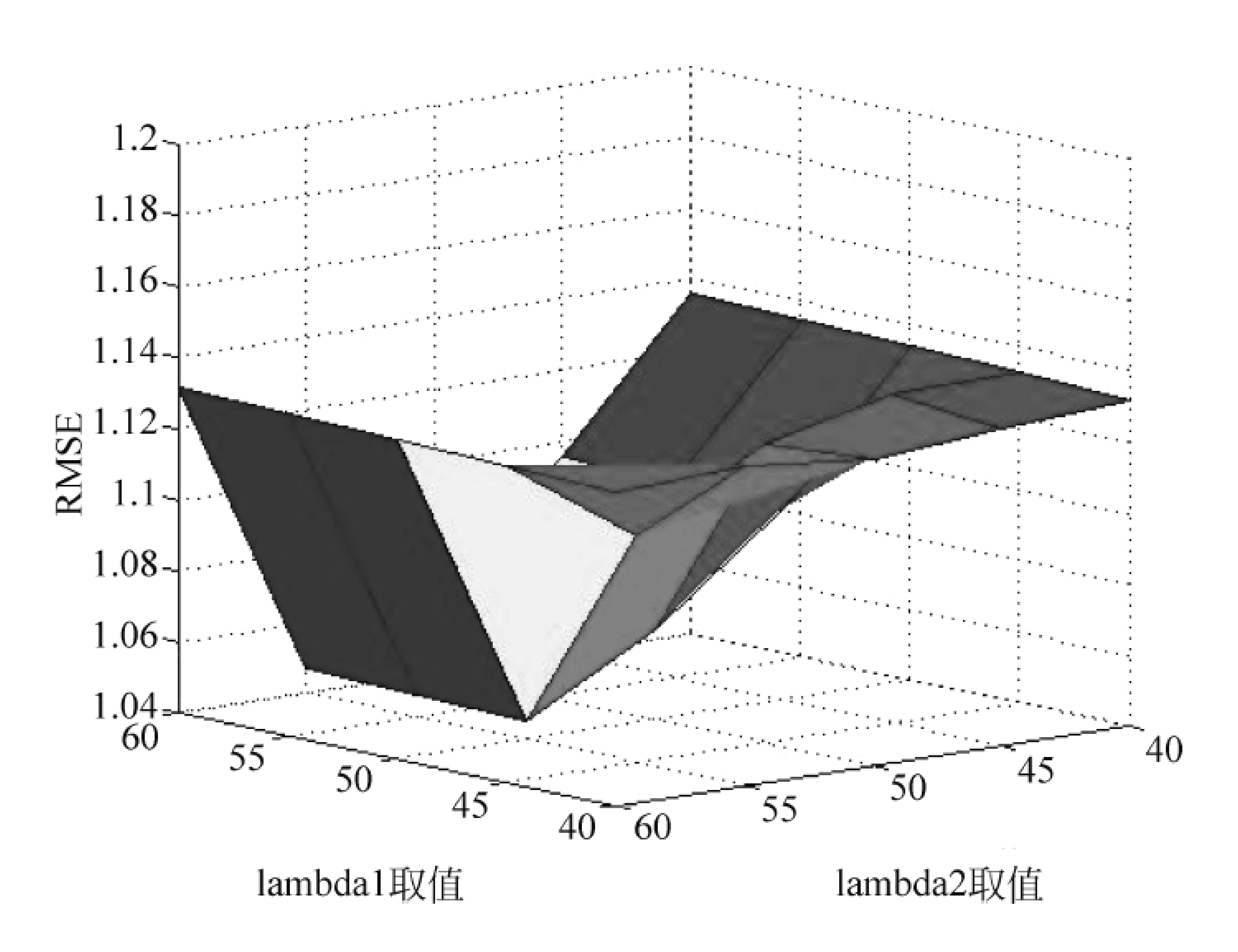


图 6-4　Epinions 数据集的 Baseline 预测

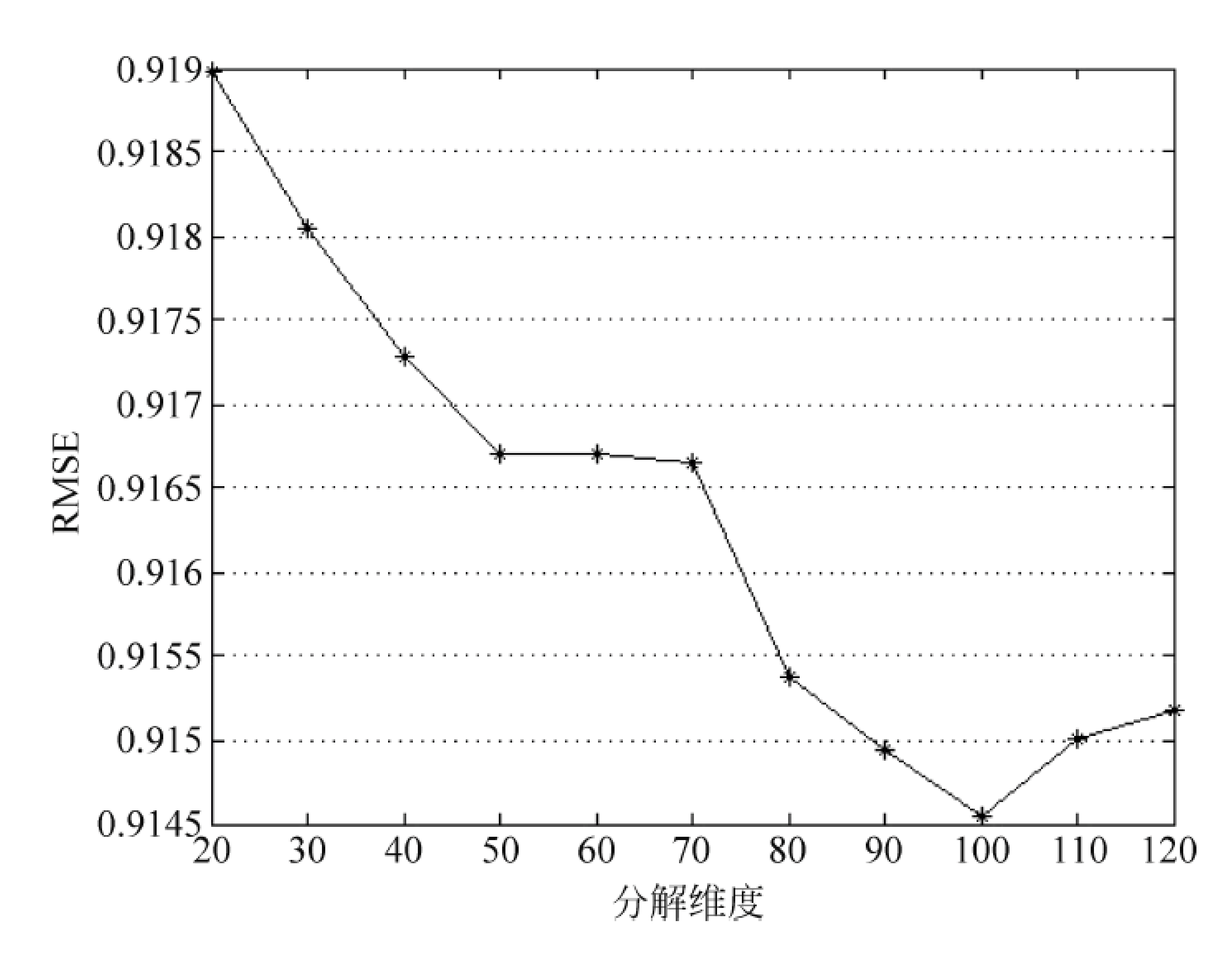


图 6-5　MovieLens 数据集潜在因子维度

通过图 6-5 至图 6-7 可以看出，BSPMF 算法对于不同的数据集，在 RMSE 最优的情况下其维度差距比较大，说明算法在推荐过程中，数据集的选择对于算法的实现起着至关重要的作用。为了更直观地表示同一算法在 RMSE 最优的情况下不同数据集的变化，绘制成如图 6-8 所示的 BSPMF 算法对 3 种数据集的 RMSE 对比图。

为了进一步观察不同算法对上述 3 种数据集的 RMSE 值，根据图 6-5 至图 6-7，选定 MovieLens 数据集的潜在因子维度为 100，Ciao 数据集的潜在因子维度为 60，Epinions 数据集潜在因子维度为 80。根据实验确定算法的正则化因子 $\lambda u=0.09$、$\lambda v=0.06$、$\lambda bu=0.09$、$\lambda bi=0.06$ 和 SGD 学习速率 alpha＝0.003，得到最终的 RMSE。如图 6-9 所示为 3 种不同数据集下不同算法的 RMSE。

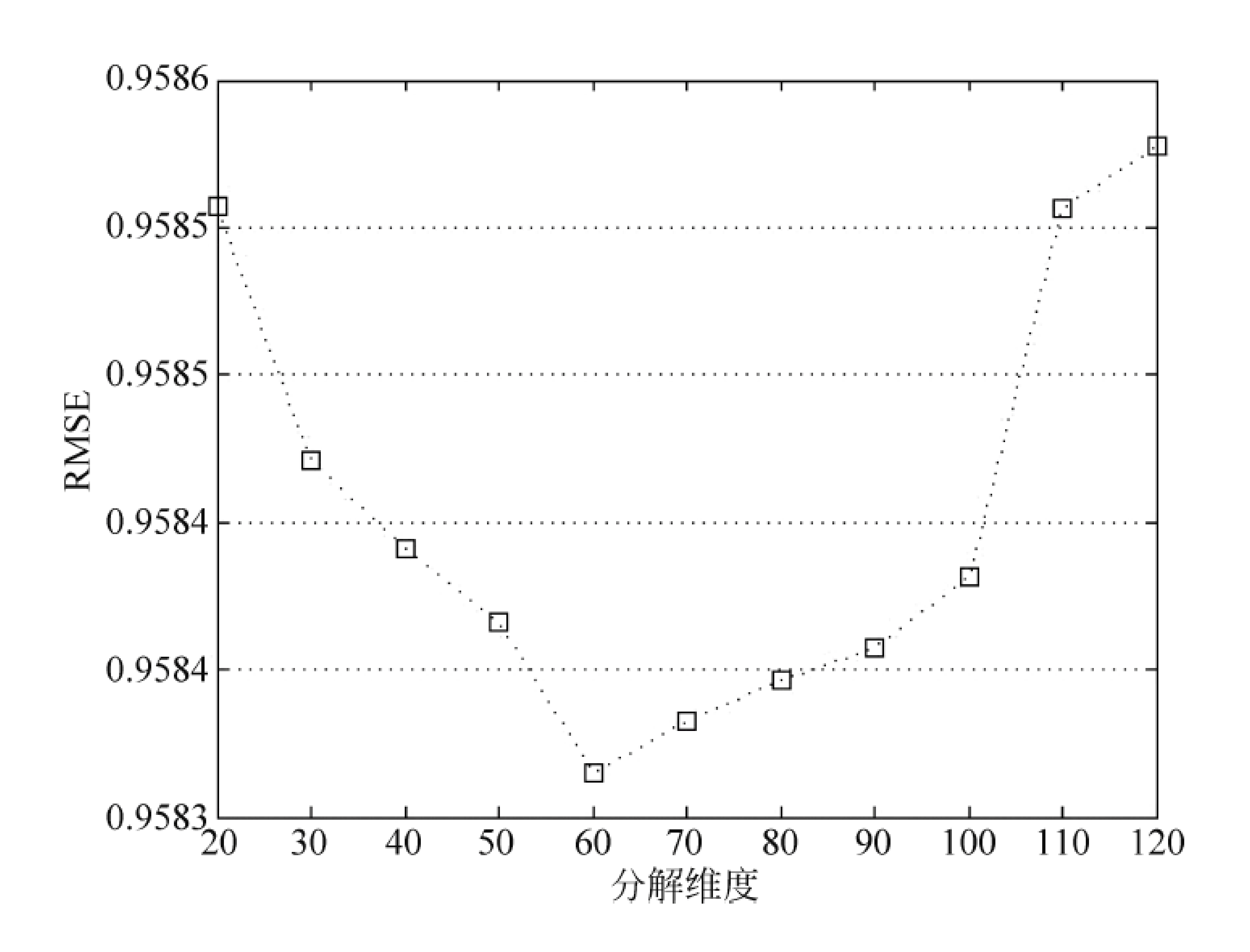


图 6-6 Ciao 数据集潜在因子维度

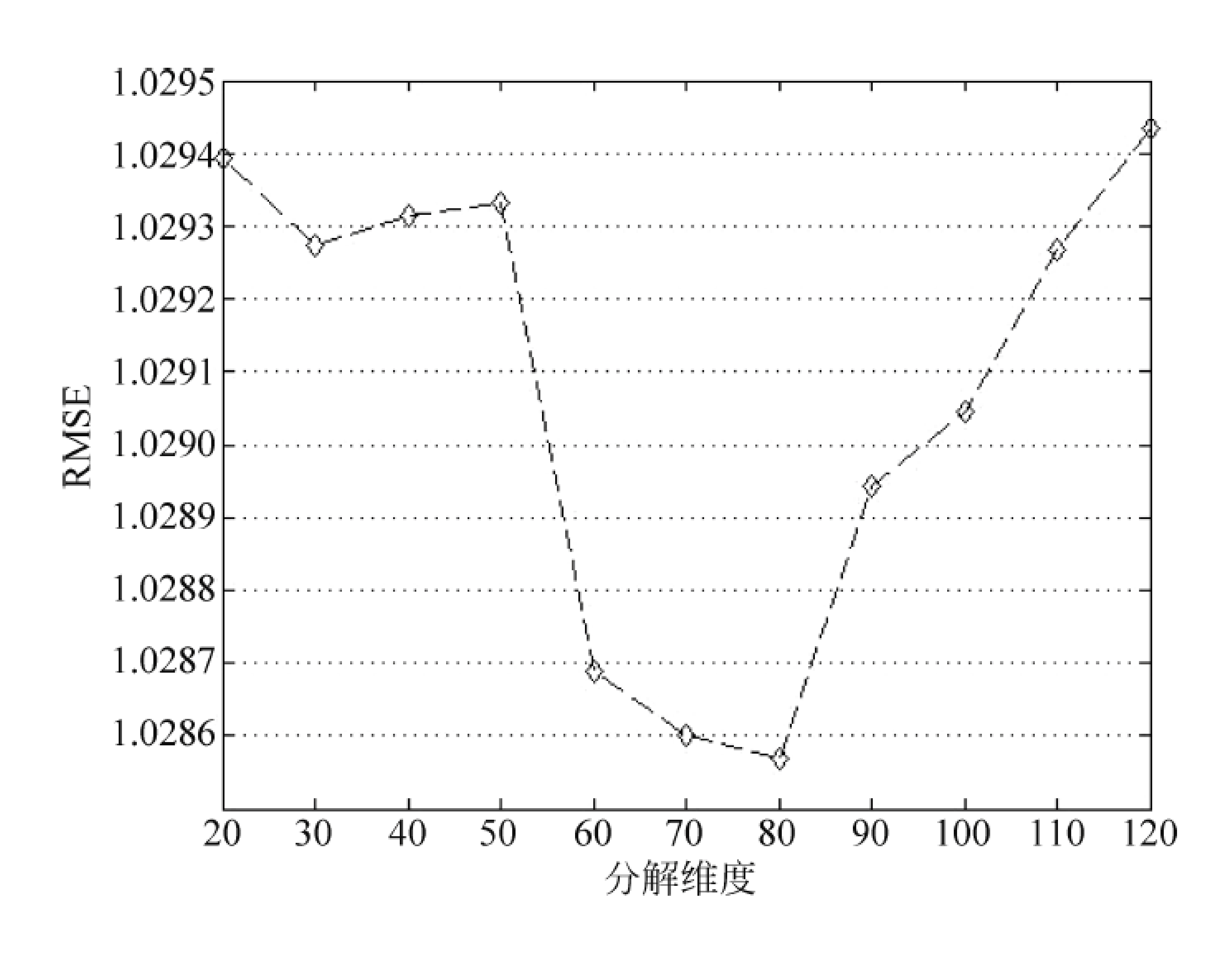


图 6-7 Epinions 数据集潜在因子维度

从图 6-9 可以看出，BSPMF 算法与经典的 userMean 算法、Baseline 预测算法、PMF 算法、BiasPMF 算法相比，在 3 种数据集下，达到 RMSE 最优情况下，BSPMF 算法效果最优。RMSE 最小值的情况与数据集本身的维度也有较大的关系，维度越大，采用 BSPMF 算法效果越明显，也进一步说明 BSPMF 算法在降低维度的同时能够缓解由数据的高维稀疏性带来的推荐精度不高的问题。

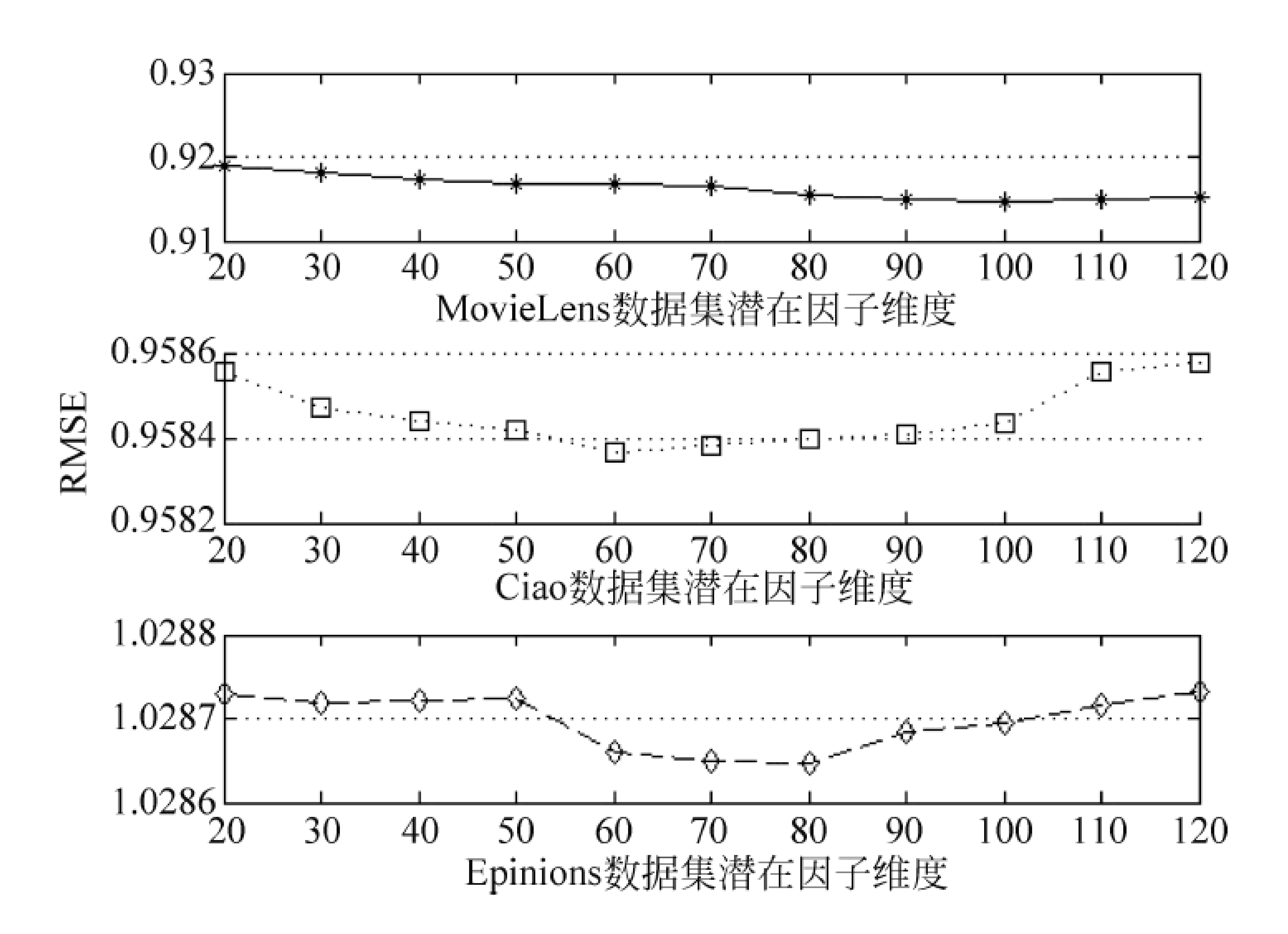


图 6-8　BSPMF 算法对 3 种数据集的 RMSE 对比图

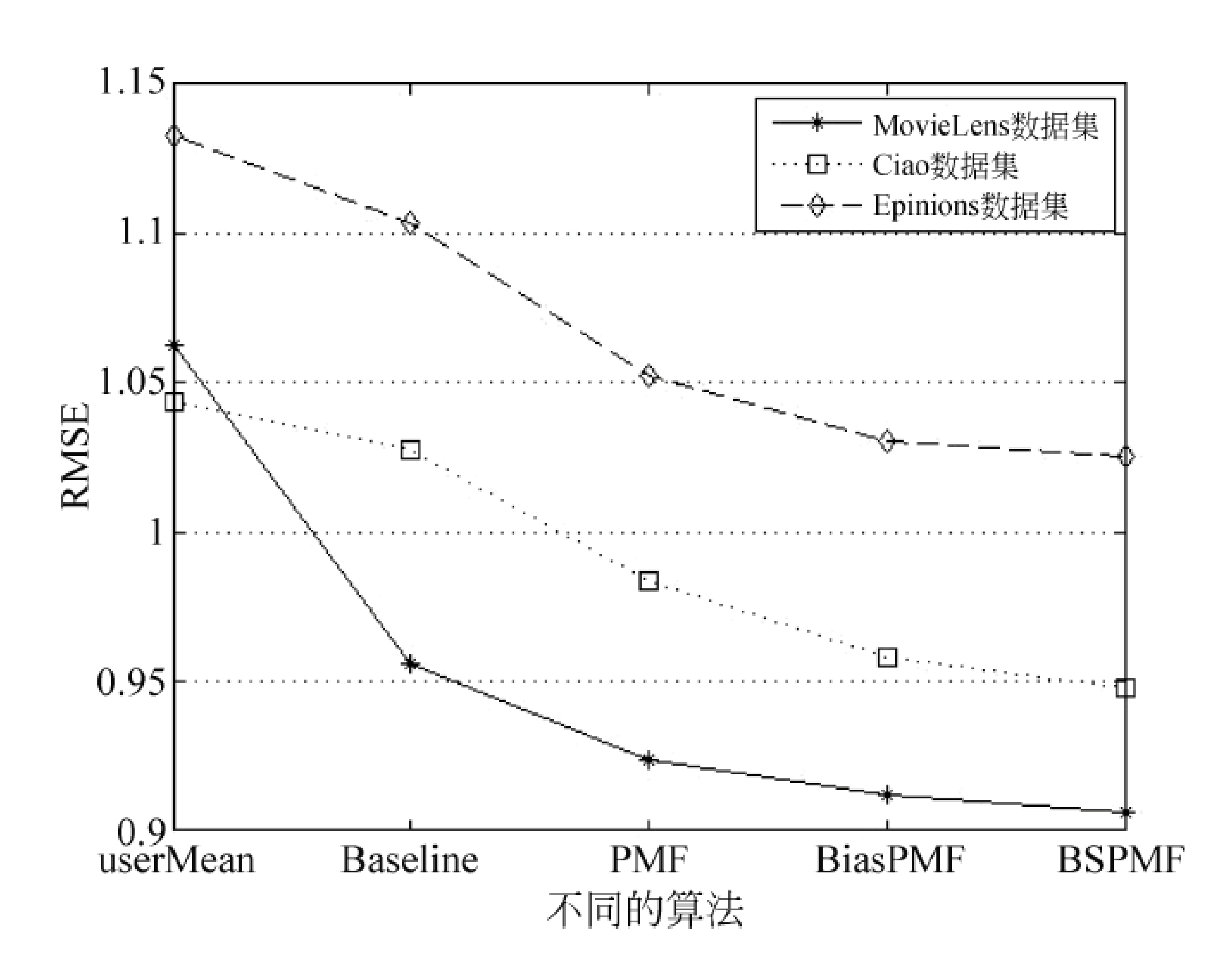


图 6-9　3 种不同数据集下不同算法的 RMSE

本章小结

本章针对个性化推荐算法所面临的由数据的高维稀疏性带来的推荐精度不高问题提出了基于偏置信息的 SVD 概率矩阵分解算法，该算法充分利用了已有的用户评分信息、偏置信息和概率矩阵分解的方法，提高了预测的精度。但是，如何整合有用的隐式反馈信息和社交网络信息，从而进一步提高推荐精度仍值得继续研究。

参考文献

[1] Lemire D, Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering[C]//SDM, 2005, 5: 1-5.

[2] 李华,张宇,孙俊华. 基于用户模糊聚类的协同过滤推荐研究[J]. 计算机科学, 2013, 39(12): 83-86.

[3] 李卫平,杨杰. 基于随机梯度矩阵分解的社会网络推荐算法[J]. 计算机应用研究, 2014, 31(6): 1654-1656.

[4] 吴湖,王永吉,王哲,等. 两阶段联合聚类协同过滤推荐算法[J]. 软件学报, 2010, 21(5): 1042-1054.

[5] Zhang R, Ooi B C, Tan K L. Making the pyramid technique robust to query types and workloads[C]//Data Engineering, 2004. Proceedings. 20th International Conference on IEEE, 2004: 313-324.

[6] 贺玲,吴玲达,蔡益朝,等. 多媒体数据挖掘中数据间的相似度度量研究[J]. 国防科技大学学报, 2006, 1.

[7] 刘庆鹏. 基于协同过滤的电子商务个性化推荐算法研究[D]. 海口: 海南大学, 2012.

[8] 曾艳,麦永浩. 一种高效的频繁模式挖掘算法[J]. 计算机应用, 2004, 24(8): 57-60.

[9] 孙小华. 协同过滤系统的稀疏性与冷启动问题研究[D]. 杭州: 浙江大学, 2005.

[10] Song Q, Cheng J, Lu H. Incremental Matrix Factorization via Feature Space Re-learning for Recommender System[C]//Proceedings of the 9th ACM Conference on Recommender Systems. ACM, 2015: 277-280.

[11] Solovyev S A, Tordeux S. An efficient truncated SVD of large matrices based on the low-rank approximation for inverse geophysical problems [J]. Сибирские электронные математические известия, 2015, 12(0): 592-609.

[12] Mnih A, Salakhutdinov R. Probabilistic matrix factorization[C]//Advances in neural information processing systems, 2007: 1257-1264.

[13] Hofmann T. Latent semantic models for collaborative filtering[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 89-115.

[14] Hastie T, Mazumder R, Lee J, et al. Matrix completion and low-rank SVD via fast alternating least squares[J]. arXiv preprint arXiv, 1410.2596, 2014.

[15] 杨阳,向阳,熊磊. 基于矩阵分解与用户近邻模型的协同过滤推荐算法[J]. 计算机应用, 2012, 32(02): 395-398.

[16] 吴扬,林世平. 基于正负反馈矩阵的 SVD 推荐模型[J]. 计算机系统应用, 2015, 24(6): 14-18.

[17] 李改,李磊. 基于矩阵分解的单类协同过滤推荐算法[J]. 计算机应用研究, 2012, 29(5): 1662-1665.

[18] Product Review Datasets: Epinions and Ciao. 亚利桑那州立大学: http://www.public.asu.edu/~jtang20/datasetcode/truststudy.htm[EB/OL],2012.10.23.

[19] MovieLens: MovieLens 100K Dataset. Grouplens: http://grouplens.org/datasets/movielens[EB/OL],2015.

[20] Lee W P, Ma C Y. Enhancing collaborative recommendation performance by combining

user preference and trust-distrust propagation in social networks[J]. Knowledge-Based Systems, 2016, 106: 125-134.

[21] Ansel M, Gauthier C. SMG: Fast scalable greedy algorithm for influence maximization in social networks[J]. Physica A Statistical Mechanics & Its Applications, 2015, 420(3): 124-133.

[22] Scott D, Dumais S T, Furnas G W, et al. Indexing by latent semantic analysis[J]. Journal of the American Society for Information Science, 1990, 41(6): 391-407.

[23] Chen H, Li Z, Hu W. An improved collaborative recommendation algorithm based on optimized usersimilarity[J]. The Journal of Supercomputing, 2016, 72(7): 2565-2578.

[24] Paterek A. Improving regularized singular value decomposition for collaborativefiltering [C]//Preceedings of KDD Cup and Workshop,California,2007,39-42.

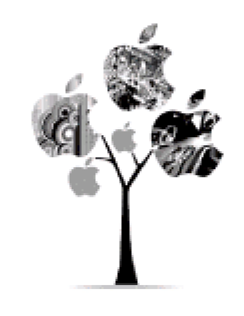
[25] Weng J, Miao C, Goh A. Improving collaborative filtering with trust-based metrics. [J]. Sac Proceedings of the Acm Symposium on Applied Computing, 2006: 1860-1864.

[26] Moradi P, Ahmadian S. A reliability-based recommendation method to improve trust-aware recommender systems[J]. Expert Systems with Applications, 2015, 42(21): 7386-7398.

[27] Wolff J G. A Scaleable Technique For Best-Match Retrieval Of Sequential Information Using Metrics-Guided Search[J]. Journal of Information Science, 1994, 20(1): 16-28.

[28] Pham X H, Nguyen T T, Jung J J. Spear: A New Method for Expert Based Recommendation Systems[J]. Journal of Cybernetics, 2014, 45(2): 165-179.

第 7 章 基于项目属性改进概率矩阵分解算法



为进一步提升推荐的精度，鉴于具有相似属性的项目之间的项目潜在因子向量也具有相似度，本章首先将相似项目的潜在因子向量的差值作为一种非线性正则项来约束传统的概率矩阵分解过程，同时考虑到同一项目有多种不同的属性，而且为避免共同属性少而相似度高的问题，提出一种拉普拉斯平滑修正的项目之间的相似度度量方法作为权重来约束项目的分解；其次融合用户项目偏置信息，提出一种名为 IAR-BP（Item Attribute Regulation with Bias Probabilistic Matrix Factorization）的改进概率矩阵分解算法。在两个真实数据集上的实验结果表明，本章提出的 IAR-BP 算法相对于传统的概率矩阵分解算法不仅收敛快而且收敛精度高。

7.1 引言

概率矩阵分解算法通过优化预先设定的目标函数从而得到近似全局最优解，假设用户—项目评分矩阵服从高斯分布，并且由其最终分解得到的用户特征矩阵、项目特征矩阵也分别服从高斯分布且两者之间相互独立同分布，推荐精度较高，同时具有坚实的理论基础，能较好地应用于实践之中。

目前大部分 PMF 算法是将用户项目—评分矩阵作为唯一的信息源，比较有代表性的有 Ruslan Salakhutdinov 等人提出的 PMF 算法，分析用户行为的隐含语义，通过隐含特征训练用户兴趣和物品。但在实际使用中很难实现实时推荐，利用时间信息对用户或项目进行建模，改进传统的隐语义模型推荐算法。Rendle 将隐语义模型的泛化表示为因子分解机 FM，针对矩阵分解容易过拟合，Salakhutdinov 等人提出贝叶斯概率矩阵分解模型 BPMF，对概率矩阵分解模型进行贝叶斯处理，并使用马尔可夫链方法训练 BPMF，所得到的模型比 PMF 提高了预测准确度。对电影评分建模时，很难将评分时间、用户属性和电影属性加入到 PMF 模型中，Adams 等人对这些信息高斯过程先验耦合再加入到 PMF 模型中，将概率矩阵分解模型与高斯过程相结合，提出了 DPMF 算法。Porteous 等人将贝叶斯概率矩阵

分解与狄利克雷过程相结合，提出了 BMFSI 算法。Koren 对隐语义模型进行了描述，并在此基础上结合用户的显示反馈和隐式反馈提出了一种称为 BiasedMF 的推荐算法，Steffen Rendle 等人利用其所提出 libFM 实现因子分解机的相关算法。国内比较有代表性的有东北大学郭贵冰副教授所领导的推荐系统团队设计开发的 libRec 推荐系统库和中国台湾师范大学林智仁教授所领导的机器学习小组研发的业内著名的 libMF 矩阵分解库并予以描述。

不过单纯依靠用户项目评分信息并不能显著提高推荐精度，因为这类方法往往假设用户和项目之间独立同分布，忽略了项目属性信息对推荐精度的影响；同时，由于算法假设用户和用户之间、项目和项目之间独立同分布，没有考虑到项目之间的复杂关系以及用户和项目本身的差异，不能显著提高推荐质量，尤其对于高维稀疏数据推荐效果往往难以得到保证。

现有的概率矩阵分解方法存在如下两大问题：

(1) 大多数前人的工作集中在额外信息和评分信息的简单线性融合，很少考虑到社交网络中信任的传播与聚合，导致推荐精度较低。

(2) 用户自定义的标签信息和评论信息往往过于随意，从而造成不精确的相似项度量，实际使用中需要结合非常复杂的自然语言处理技巧，而且并不是所有的数据集都包含用户对项目的标签信息或者评分信息。

项目属性信息一般由领域内专家进行注解，具有一定的权威性并且不需要复杂的自然语言处理技巧，例如如果某一用户喜欢看恐怖片，某部电影属于恐怖片，那么该用户就很可能喜欢这种电影，具体到本章的处理中就是给恐怖片较大的权重。文献[1]结合项目属性信息和概率矩阵分解模型提出一种名为 IAPMF 的算法来解决协同过滤中的冷启动(Cold Start)问题。文献[2]探索了利用项目信息和矩阵分解模型来解决协同过滤中的隐私保护问题的可能性。以上这些基于项目属性的方法虽然从模拟现实世界的角度出发，但它们的推荐过程将项目属性共同对待，而忽略了项目属性的多样性；然而在很多现实场景中，具有共同某些属性的项目不一定被同样的用户所喜欢，例如有的用户仅仅是出于心理的原因才看同一名演员所出演的电影；此外，现实世界中属性标注的随意性也是造成属性关系多样性的重要原因之一，例如有的电影是动作类喜剧，但是标注为搞笑类或者仅仅是动作类，因此简单地将属性关系同等对待，认为喜欢动作类电影的用户一定喜欢另外的具有动作属性的电影，将很难发现隐藏在项目属性背后的项目关联关系。

7.2　IAR-BP 算法

7.2.1　相似度度量

传统的项目属性相似度度量方法仅仅依靠类似 Jaccard 系数的方式或者用用户对项目的评分相似度作为度量标准，这在一定程度上缓解了由数据稀疏性带来

的推荐精度不高的问题，但是该方法存在着以下的固有缺陷和不足。

(1) 每个项目的属性都是多样的，所以很难用某一属性进行概述。尽管用户对项目的评价很高，但是用户有可能喜好评价项目赋予的某些属性。所以，单纯地通过用户对项目的评价来判断用户间是否存在共同喜好具有一定的片面性，无法从本质上有效地反映用户的真实喜好。

(2) 如果采用 Jaccard 系数，那么会出现该方法避免了一些公共项目数目少而相似度值高的不合理现象，例如假设某类项目有 200 个属性，如果 i_1 和 i_2 共同有 2 个而且只有 2 个，同样 i_3 和 i_4 共同有 20 个而且只有 20 个，那么通过它们的 Jaccard 系数得到的相似度最终都是 1。

(3) 这些方法没有考虑属性之间的区分度，例如判断男人和女人，如果某用户有很多胡子，基本就是男人了。而且可能一些属性区分度不是很大，例如电影类型(爱情、动作、喜剧)，这些属性并不是互斥的。

鉴于此，本章提出相似度度量方式如式(7-1)所示。

$$\text{itemsim}(i,j)=\frac{\sum_{t\in N(it\text{Set}\cap jt\text{Set})}W_t+1}{\sum_{t\in N(it\text{Set}\cup jt\text{Set})}W_t+k}*\frac{N(it\text{Set}\cap jt\text{Set})+1}{N(it\text{Set}\cup jt\text{Set})+k} \tag{7-1}$$

式中：$it\text{Set}$——电影 i 具有的项目属性集合；

$N(it\text{Set}\cap jt\text{Set})$——电影 i 和 j 具有的项目属性的交集；

$N(it\text{Set}\cup jt\text{Set})$——并集；

k——拉普拉斯平滑；

w——权重系数，来自于该属性类电影被评价的次数除以总的次数，最后做归一化。

7.2.2 算法描述

数据集包括一个含有 N 个用户的用户集合 $U=\{u_1,u_2,u_3,\cdots,u_N\}$ 和 M 个项目的集合 $I=\{i_1,i_2,i_3,\cdots,i_M\}$，用户项目评分矩阵用 R_{NM} 表示。

IAR-BP 算法的基本思想是在矩阵分解的基础上引入概率的思想，首先假设用户对项目的真实评分和预测评分之间的误差服从高斯分布，因此观察到的用户评分服从以下的条件分布，如式(7-2)所示。

$$R_{ij}\leftarrow U_i^{\text{T}}V_j\Rightarrow p(R_{ij}-U_i^{\text{T}}V_j\mid 0,\sigma^2)\Leftrightarrow p(R_{ij}\mid U_i^{\text{T}}V_j,\sigma^2) \tag{7-2}$$

然后假设用户潜在因子空间服从均值为 0 的高斯分布，潜在因子之间独立同分布，如式(7-3)所示。

$$p(U\mid\sigma_U^2)=\prod_{i=1}^{N}[N(U_i\mid 0,\sigma_U^2)] \tag{7-3}$$

式中：$N(x|\mu,\sigma^2)$——均值为 μ、方差为 σ^2 的高斯分布。

其次，由于项目属性的影响，越相似的项目之间的项目潜在因子空间也越相似，也就是说用一个项目的潜在因子空间需要根据和其相似的其他项目的潜在因

子空间来决定，根据概率论中的链式法则，项目的潜在因子空间如式(7-4)所示。

$$p(V \mid \text{itemsim}, \sigma_V^2, \sigma_{\text{itemsim}}^2) \propto p(V \mid \sigma_V^2) \times p(V \mid \text{itemsim}, \sigma_{\text{itemsim}}^2)$$
$$= \prod_{j=1}^{M} [N(V_j \mid 0, \sigma_V^2)] \times \prod_{j=1}^{M} \left[N\left(V_j \middle| \sum\nolimits_{t \in V} \text{itemsim}(j,t) V_t, \sigma_{\text{itemsim}}^2\right)\right] \quad (7\text{-}4)$$

对 R_{NM} 经过贝叶斯推导，隐式特征的后验概率如式(7-5)所示。

$$p(U, V \mid R, \text{itemsim}, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{\text{itemsim}}^2) \propto$$
$$p(R \mid U, V, \text{itemsim}, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{\text{itemsim}}^2) \times p(U, V, \text{itemsim}, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{\text{itemsim}}^2)$$
$$= p(R \mid U, V, \sigma^2) \times p(U \mid \sigma_U^2) \times p(V \mid \text{itemsim}, \sigma_V^2, \sigma_{\text{itemsim}}^2)$$
$$= p(R \mid U, V, \sigma^2) \times p(U \mid \sigma_U^2) \times p(V \mid \sigma_V^2) \times p(V \mid \text{itemsim}, \sigma_{\text{itemsim}}^2)$$
$$= \prod_{i=1}^{N} \prod_{j=1}^{M} [N(R_{ij} \mid U_i^{\mathrm{T}} V_j, \sigma^2)]^{I_{ij}} \times \prod_{i=1}^{N} N(U_i \mid 0, \sigma_U^2) \times \prod_{j=1}^{M} N(V_j \mid 0, \sigma_V^2) \times$$
$$\prod_{j=1}^{M} \left[N\left(V_j \middle| \sum_{t \in V} \text{itemsim}(j,t) V_t, \sigma_{\text{itemsim}}^2\right)\right] \quad (7\text{-}5)$$

通过最大似然估计来最大化后验概率，如式(7-6)所示。

$$p(U, V \mid R, \text{itemsim}, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{\text{itemsim}}^2)$$
$$= -\frac{1}{2\sigma^2} \prod_{i=1}^{N} \prod_{j=1}^{M} I_{ij} (R_{ij} - U_i^{\mathrm{T}} V_j)^2 - \frac{1}{2\sigma_U^2} \sum_{i=1}^{N} U_i^{\mathrm{T}} U_i -$$
$$\frac{1}{2\sigma_V^2} \sum_{j=1}^{M} V_j^{\mathrm{T}} V_j - \frac{1}{2} \left(\sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} \right) \ln \sigma^2 -$$
$$\frac{1}{2\sigma_{\text{itemsim}}^2} \sum_{j=1}^{M} \left(V_j - \sum\nolimits_{t \in V} \text{itemsim}(j,t)\right)^{\mathrm{T}} \left(V_j - \sum\nolimits_{t \in V} \text{itemsim}(j,t)\right) -$$
$$\frac{1}{2}((N \times K) \ln \sigma_U^2 + (M \times K) \ln \sigma_V^2 +$$
$$(M \times K) \ln \sigma_{\text{itemsim}}^2) + C \quad (7\text{-}6)$$

式中：C——常量。

C 通过固定超参数 $\sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{\text{itemsim}}^2$，如式(7-7)所示。

$$L(R, U, V, \text{itemsim}) = \prod_{i=1}^{N} \prod_{j=1}^{M} I_{ij} (R_{ij} - U_i^{\mathrm{T}} V_j)^2 + \lambda_U \| U \|_F^2 + \lambda_V \| V \|_F^2 +$$
$$\lambda_{\text{itemsim}} \sum_{j=1}^{M} \left(V_j - \sum\nolimits_{t \in V} \text{itemsim}(j,t) V_t\right)^{\mathrm{T}}$$
$$\left(V_j - \sum\nolimits_{t \in V} \text{itemsim}(j,t) V_t\right) \quad (7\text{-}7)$$

式中：$\lambda_U = \dfrac{\sigma^2}{\sigma_U^2}$、$\lambda_V = \dfrac{\sigma^2}{\sigma_V^2}$、$\lambda_{\text{itemsim}}$——控制着项目属性信息在矩阵分解中的作用；

$\| * \| F$——矩阵的 F 范数，求导得到各个变量的偏导。

例如：要预测用户 user1 对电影 movie1 的打分，假设电影数据集总体偏置 μ 为3.4 分，电影 movie1 的口碑比其他电影高 0.9 分，即 bi(movie1)=0.9，另一方面用户 user1 是一个乐观的用户，一般偏向于给电影打低 0.3 分，即 bu(user1)=

−0.3，那么用户 user1 对电影 movie1 的预测打分为 3.4+0.9−0.3=4 分。

为了不失一般性，往往用 $f(x)=(x-1)/(R_{\max}-1)$ 把实际评分映射到(0,1]，用 $g(x)=1/(1+\exp(x))$ 把预测评分映射到(0,1]，其中 $R_{\max}$ 为数据集中的最大评分值，如式(7-8)所示。

$$\begin{aligned} L(R,U,V,\text{itemsim}) = & \prod_{i=1}^{N}\prod_{j=1}^{M} I_{ij}(R_{ij} - g(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V_j))^2 + \\ & \lambda_U \|U\|_F^2 + \lambda_V \|V\|_F^2 + \lambda_{bu}\|bu\|_F^2 + \lambda_{bi}\|bi\|_F^2 + \\ & \lambda_{\text{itemsim}} \sum_{j=1}^{M} \left(V_j - \sum\nolimits_{t\in V} \text{itemsim}(j,t)V_t\right)^{\mathrm{T}} \\ & \left(V_j - \sum\nolimits_{t\in V} \text{itemsim}(j,t)V_t\right) \end{aligned} \tag{7-8}$$

最终的目标函数如式(7-9)所示。

$$\begin{aligned} L(R,U,V,\text{itemsim}) = & \prod_{i=1}^{N}\prod_{j=1}^{M} I_{ij}(R_{ij} - g(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V_j))^2 + \\ & \lambda_U \|U\|_F^2 + \lambda_V \|V\|_F^2 + \lambda_{bu}\|bu\|_F^2 + \lambda_{bi}\|bi\|_F^2 + \\ & \lambda_{\text{itemsim}} \sum_{j=1}^{M} \left(V_j - \sum\nolimits_{t\in V} \text{itemsim}(j,t)V_t\right)^{\mathrm{T}} \\ & \left(V_j - \sum\nolimits_{t\in V} \text{itemsim}(j,t)V_t\right) \end{aligned} \tag{7-9}$$

式中：μ、U、V 需要学习，对应的梯度如式(7-10)所示。

$$\begin{cases} \nabla U_i = \dfrac{\partial L}{\partial U_i} = \displaystyle\sum_{j=1}^{M} I_{ij}^{R} g'(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V) \\ \qquad (g(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V_j) - R_{ij})V_j + \lambda_U U_i \\ \nabla V_j = \dfrac{\partial L}{\partial V_j} = \displaystyle\sum_{i=1}^{N} I_{ij}^{R} g'(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V) \\ \qquad (g(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V_j) - R_{ij})U_i + \\ \qquad \lambda_{\text{itemsim}} \displaystyle\sum_{i=1}^{N}\sum_{j=1}^{M} \text{itemsim}(i,j)(V_i - V_j) + \lambda_V V_j \\ \nabla bu(i) = \dfrac{\partial L}{\partial bu(i)} = \displaystyle\sum_{i=1}^{N} I_{ij}^{R} g'(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V) \\ \qquad (g(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V_j) - R_{ij}) + \lambda_{bu} bu(i) \\ \nabla bi(j) = \dfrac{\partial L}{\partial bi(j)} = \displaystyle\sum_{i=1}^{N} I_{ij}^{R} g'(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V) \\ \qquad (g(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V_j) - R_{ij}) + \lambda_{bi} bi(j) \\ \nabla \mu = \dfrac{\partial L}{\partial \mu} = \displaystyle\sum_{i=1}^{N} I_{ij}^{R} g'(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V) \\ \qquad (g(\mu + bu(i) + bi(j) + U_i^{\mathrm{T}}V_j) - R_{ij}) \end{cases} \tag{7-10}$$

然后根据式(7-11)更新参数上述参数。

$$\begin{cases} U_i \leftarrow U_i - \gamma \nabla U_i \\ V_j \leftarrow V_j - \gamma \nabla V_j \\ bu(i) \leftarrow bu(i) - \gamma \nabla bu(i) \\ bi(j) \leftarrow bi(j) - \gamma \nabla bi(j) \\ \mu \leftarrow \mu - \gamma \nabla \mu \end{cases} \tag{7-11}$$

式中：$\gamma>0$——学习速率；

μ——数据集的总体偏置；

bu——各个用户的偏置向量；

bi——各个项目的偏置向量。

最终的预测公式如式(7-12)所示。

$$\hat{R}_{ij} = \mu + bu(i) + bi(j) + U_i^{\mathrm{T}} V_j \tag{7-12}$$

图 7-1 为 IAR-BP 算法的概率图模型。

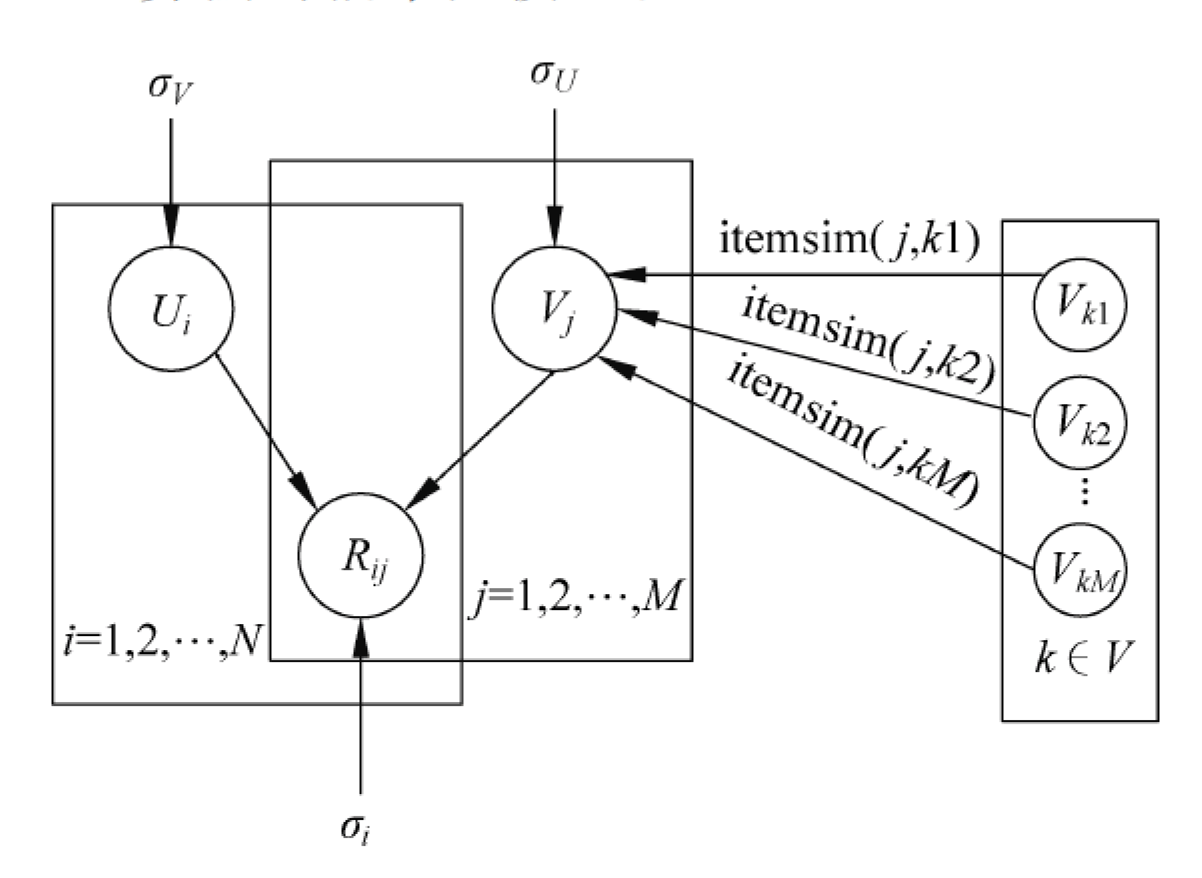


图 7-1　IAR-BP 算法的概率图模型

算法实现的伪代码如算法 7-1 所示。

算法 7-1　IAR-BP 算法

输入：maxIter，γ，p，正则化参数分解维度
输出：U，V，bu，bi，μ
算法的基本流程如下：

```
Initialize model parameters U, V, bu, bi, μ
for i=1,2,…,maxIter do
  for h=1,2,…,p do
    Randomly draw a rating record (u,i,r_ui)
    Caculate the gradients as shown in 式(7-9)
    Update model parameters as shown in 式(7-10)
  end for
  Decrease the learning rate γ←γ×0.9
end for
算法结束
```

7.2.3 算法复杂度分析

对式(7-9)进行计算的复杂度为$O(\rho_R k+mu_T d)$，ρ_R为R中已评分元素的个数，u_T为一个用户平均信任的用户个数。梯度下降的复杂度$O(\rho_R k^2+m(u_T+u'_T)u_T d)$、$O(\rho_R k^2)$、$O(\rho_R k^2)$、$O(\rho_R k^2)$和$O(\rho_R k^2)$，其中$u'_T$表示一个用户平均被信任的用户个数。

由于用户在互联网中评分记录和信任记录服从幂律分布，长尾上的用户往往只有很少的用户信任数目，那么$mu_T \ll \rho_R$；同理，$mu'_T \ll \rho_R$，那么算法最终的复杂度为$O(\rho_R k+5\rho_R k^2)$，可以看出本章提出的RBPT算法和ρ_R线性相关，适用于大数据集。

7.3 实验结果对比分析

本节通过详尽的实验来进一步地验证7.2节中提出IAE-BP算法的推荐精度，同时讨论有关维度和$\lambda_{itemsim}$的系数变化对推荐精度的影响程度，最后将本章提出的IAR-BP推荐算法与其他传统的基于矩阵分解的个性化推荐算法进行全方位的比较和分析。

7.3.1 实验数据集

在推荐系统的研究中几种公开可用的数据集被用来评价推荐算法的质量，例如MovieLens 100k、MovieLens 1M、Epinions和Ciao等人，本章选择MovieLens 100k和MovieLens 1M作为实验数据集。Ciao和Epinions包含项目属性信息，Epinions数据集包括49290用户对139738项目的评分，以及487181条用户间的信任关系。为了更加精确地验证本章提出算法的优劣，对于上述整个实验数据集需要进一步划分为训练集和测试集两部分，即按照通用的数据集和训练集的划分方法，将整个数据集的80%作为训练集，20%作为测试集。表7-1所列为MovieLens数据集中不同电影类型的属性表，数据集为MovieLens100k和MovieLens1M数据集中所包含的18种项目属性信息。从表7-1可以看出，属性之间并不是互斥的，例如Action(动作系列)和Adventure(冒险系列)等。

表7-1 MovieLens数据集中不同电影类型的属性表

Action	Adventure	Animation	Children's	Comedy	Crime
Documentary	Drama	Fantasy	Film-Noir	Horror	Musical
Mystery	Romance	Sci-Fi	Thriller	War	Western

7.3.2 实验评价标准

为检验本章提出算法的推荐质量，实验采用RMSE。

7.3.3　对比实验配置及说明

1. Baseline 预测

如式(7-2)所示是 Baseline 算法的最终评分预测公式，用该算法做对比实验的目的是比较偏置对实验结果的影响，如式(7-13)所示。

$$\hat{R}_{ij} = \mu + bu(i) + bi(j) \tag{7-13}$$

式中：$\hat{R}_{ij}$——用户 i 对项目 j 的预测评分；

μ——数据集的总体偏置信息；

$bu(i)$——用户 i 的偏置信息；

$bi(j)$——项目 j 的偏置信息。

在实际应用中往往根据经验似然采用式(7-14)的方法来求解 bi 和 bu 的值。

$$bi = \frac{\sum_{u \in R(i)} (R_{ui} - \mu)}{\alpha + |R(i)|}$$

$$bu = \frac{\sum_{i \in R(u)} (R_{ui} - \mu - bi)}{\beta + |R(u)|} \tag{7-14}$$

式中：u——某一用户；

i——某一项目；

$R(i)$——评价过项目 i 的用户集合；

$R(u)$——用户 u 评价过项目集合；

α 和 β——压缩系数，需要实验确定。

2. SVD

与传统的 SVD 算法进行对比，这里只计算用户指定的最大的 K 个奇异值。

3. PMF

实验中为了降低模型的复杂度，本章选择 $\lambda_U = \lambda_V = 0.01$，梯度下降的学习速率 $\eta = 0.03$。

4. ALS 算法

随机选择 90%作为训练集，余下的 10%作为训练集，做五折交叉验证取最终的平均值作为结果。不失一般性，对于矩阵分解算法，$\lambda_U = \lambda_V = \lambda_S = 0.1$，$\eta = 0.005$。迭代次数为 200 次。

7.3.4　实验参数分析

1. λ 对 RMSE 的影响

如图 7-2 所示为 RMSE 值的变化(100k 数据集)，在 MovieLens 100k 数据集上的 Baseline 预测，通过实验发现，当 $\lambda_1 = 2$，$\lambda_2 = 5$ 时 RMSE 达到最优，最小值为 0.9559。

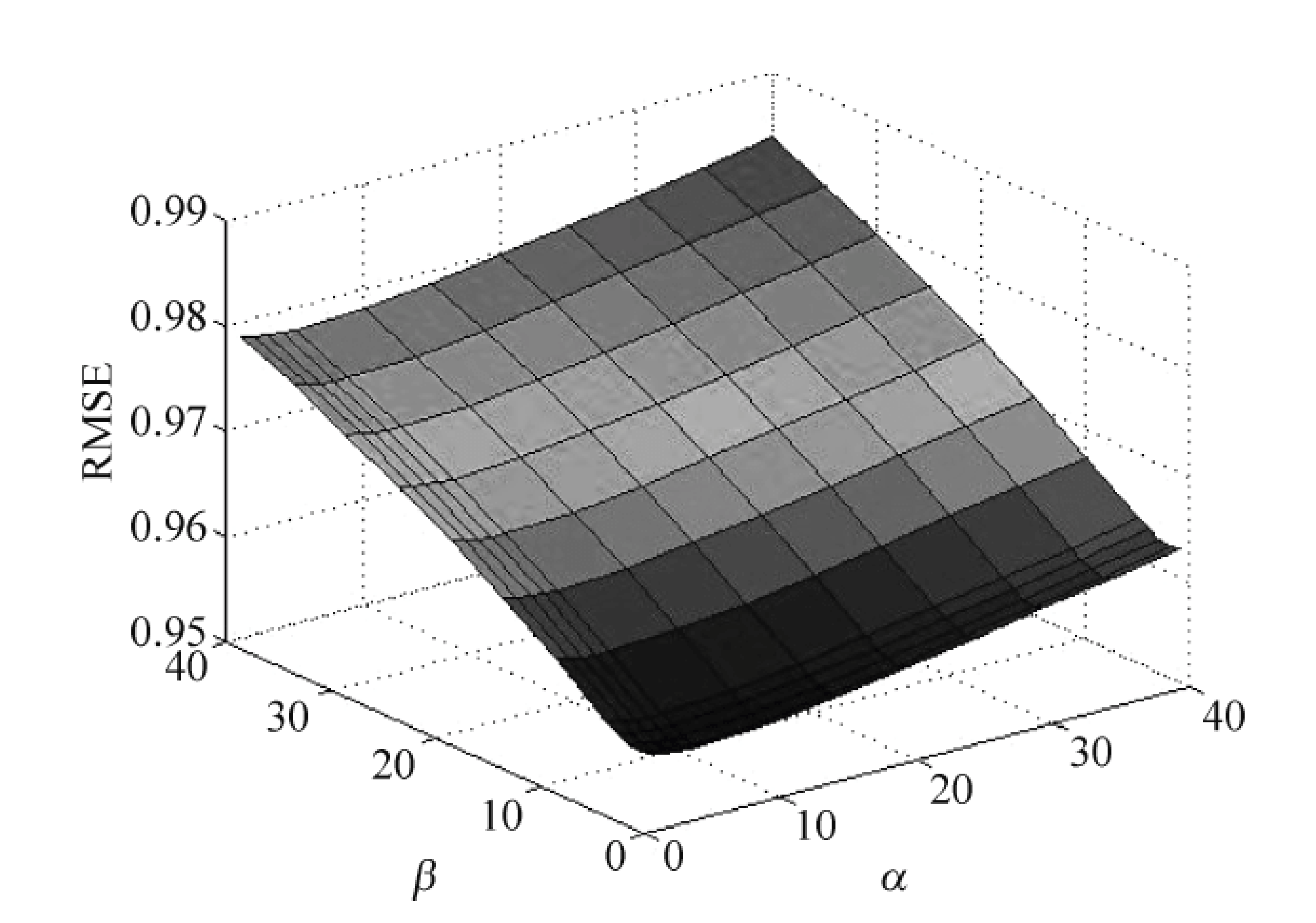


图 7-2 RMSE 值的变化(100k 数据集)

如图 7-3 所示为 RMSE 值的变化(1M 数据集),在 MovieLens 1M 数据集上做的 Baseline 预测,当 $\lambda_1=2,\lambda_2=5$ 时 RMSE 达到最优,Baseline 得到的最优值为 0.9553。

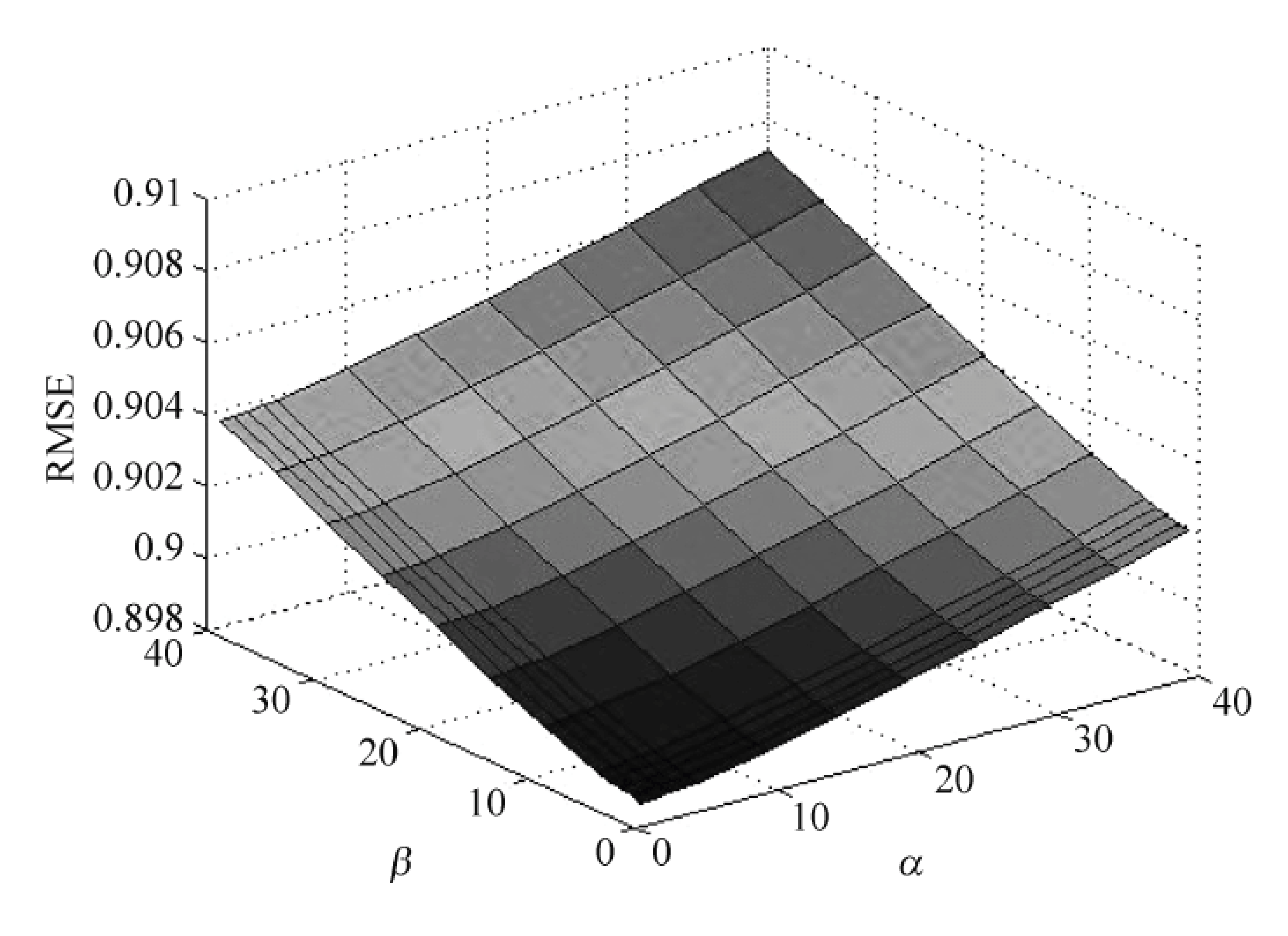


图 7-3 RMSE 值的变化(1M 数据集)

2. β 的影响

本章算法的主要优势便是联合项目属性信息和用户项目评分矩阵信息,从而显著预测用户偏好。

本章算法中的参数 β 控制着项目属性信息的贡献大小,如果 $\beta=0$,本章算法仅仅挖掘用户项目评分矩阵信息;另外,如果 $\beta=\text{inf}$,那么项目属性信息将会主宰模型的训练过程。因此在实际应用中需要精确调整参数 β 以避免模型陷入极端。如图 7-4 所示为 Ciao 数据集 β 对 MSE 值的影响,如图 7-5 所示为 FilmTrust 数据集 β 对 MSE 值的影响。

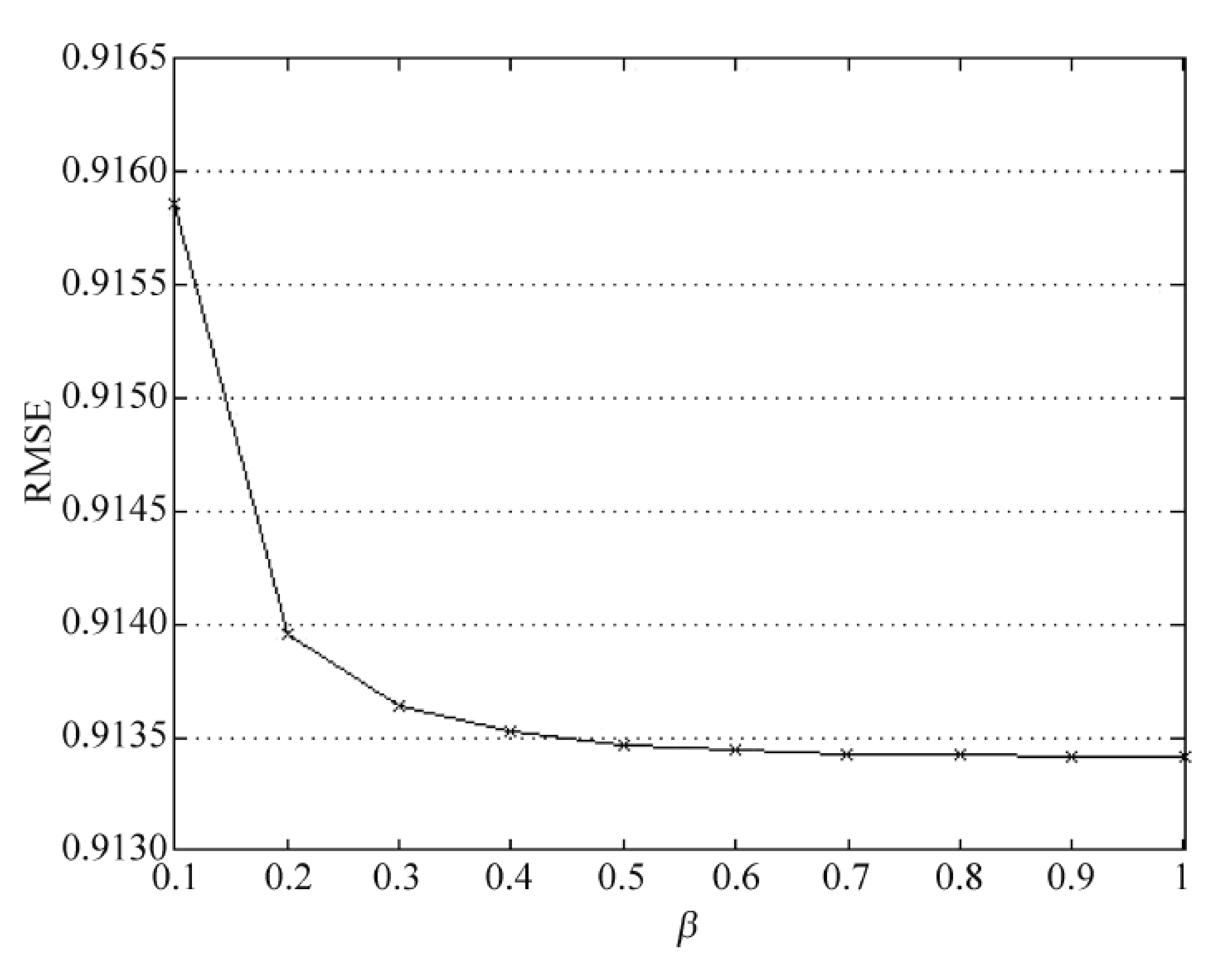


图 7-4 Ciao 数据集 β 对 MSE 值的影响

图 7-4 中 Ciao 数据集随着迭代次数的增加，RMSE 值一直处于下降过程中，尤其是一开始时，当 β 值由 0.1 到 0.2 这个变化阶段，RMSE 值呈现指数级的下降，说明 β 值对于 RMSE 的影响比较明显。当 β 值 0.7 时，RMSE 值为 0.91348。

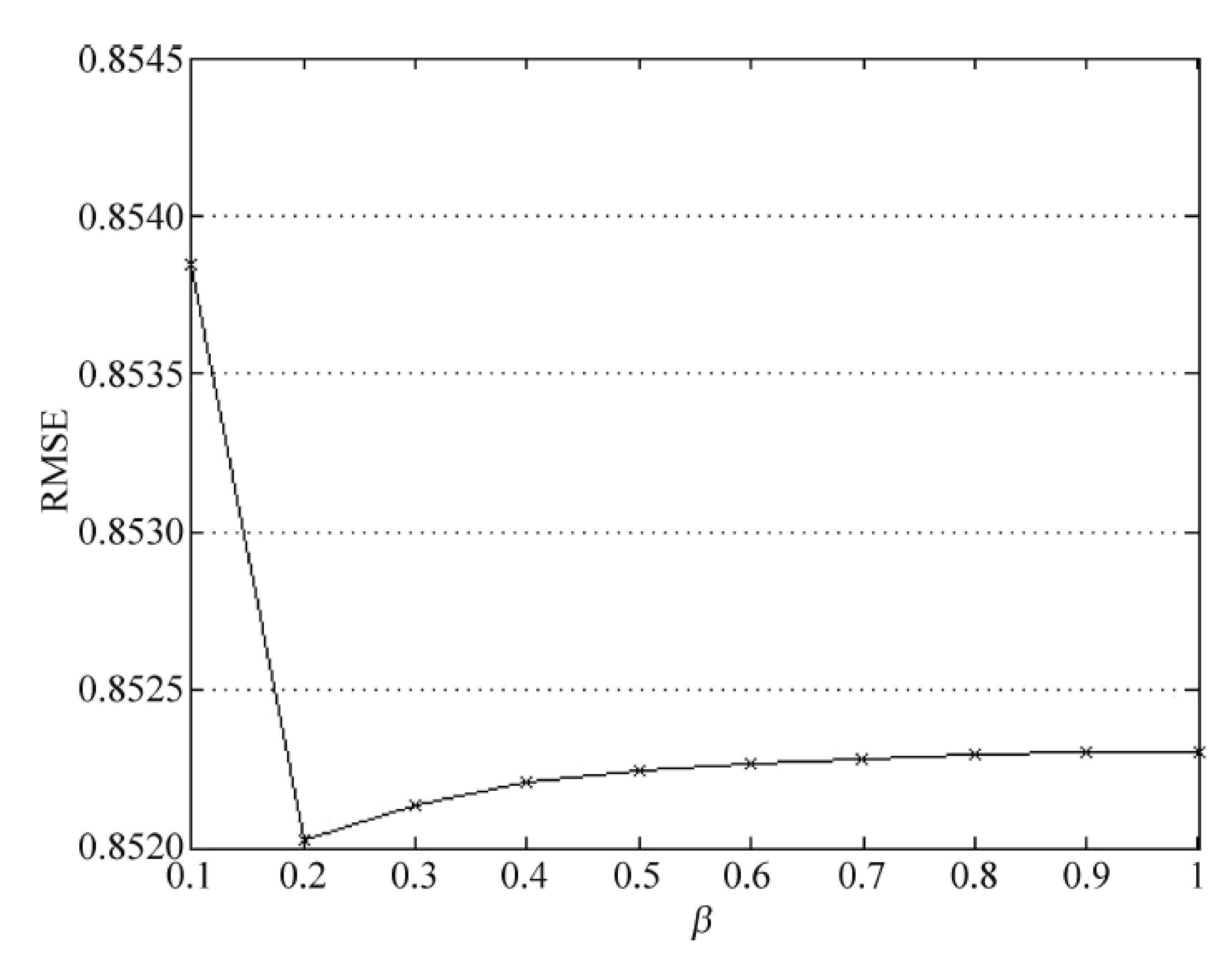


图 7-5 FilmTrust 数据集 β 对 MSE 值的影响

图 7-5 中随着 β 的增加 RMSE 先下降，当 β 超过特定值时 RMSE 又开始上升，这种现象表明融合项目属性信息能提升推荐质量。值得注意的是，在 MovieLens 100k 中当 β 超过 0.3 时，RMSE 缓慢下降，在 1M 数据集中超过 0.2 时，RMSE 缓慢上升，β 改变显著说明模型已参与训练。

从图 7-4 和图 7-5 中 β 对 RMSE 的影响，可观察到 β 显著影响推荐质量，说明融合属性信息具有显著优势。

7.3.5 实验对比

为了验证 IAR-BP 算法的有效性，本章在两个真实数据集 MovieLens 100k 和 MovieLens 1M 上做了几组实验，将 IAR-BP 算法和其他算法进行了对比，主要解决如下几个问题：

(1) 和其他协同过滤推荐算法的比较；

(2) 迭代次数对 RMSE 值的影响。

如图 7-6 所示为 MovieLens 100k 不同算法的 RMSE 值对比，如图 7-7 所示为 Movielens 1M 不同算法的 RMSE 值对比。从图 7-6 和图 7-7 可以看出，IAR-BP 算法在迭代 80 次时趋于 RMSE 值开始优于 T-SVD 算法，而且随着迭代单次数的继续增加，RMSE 值还在减小，直到迭代次数 160 左右达到最优值 0.932 并稳定下来。

从图 7-6 和图 7-7 可以看出，在数据集中，IAR-BP 算法的 RMSE 值到迭代次数 140 左右达到最优值 0.862 并稳定下来。

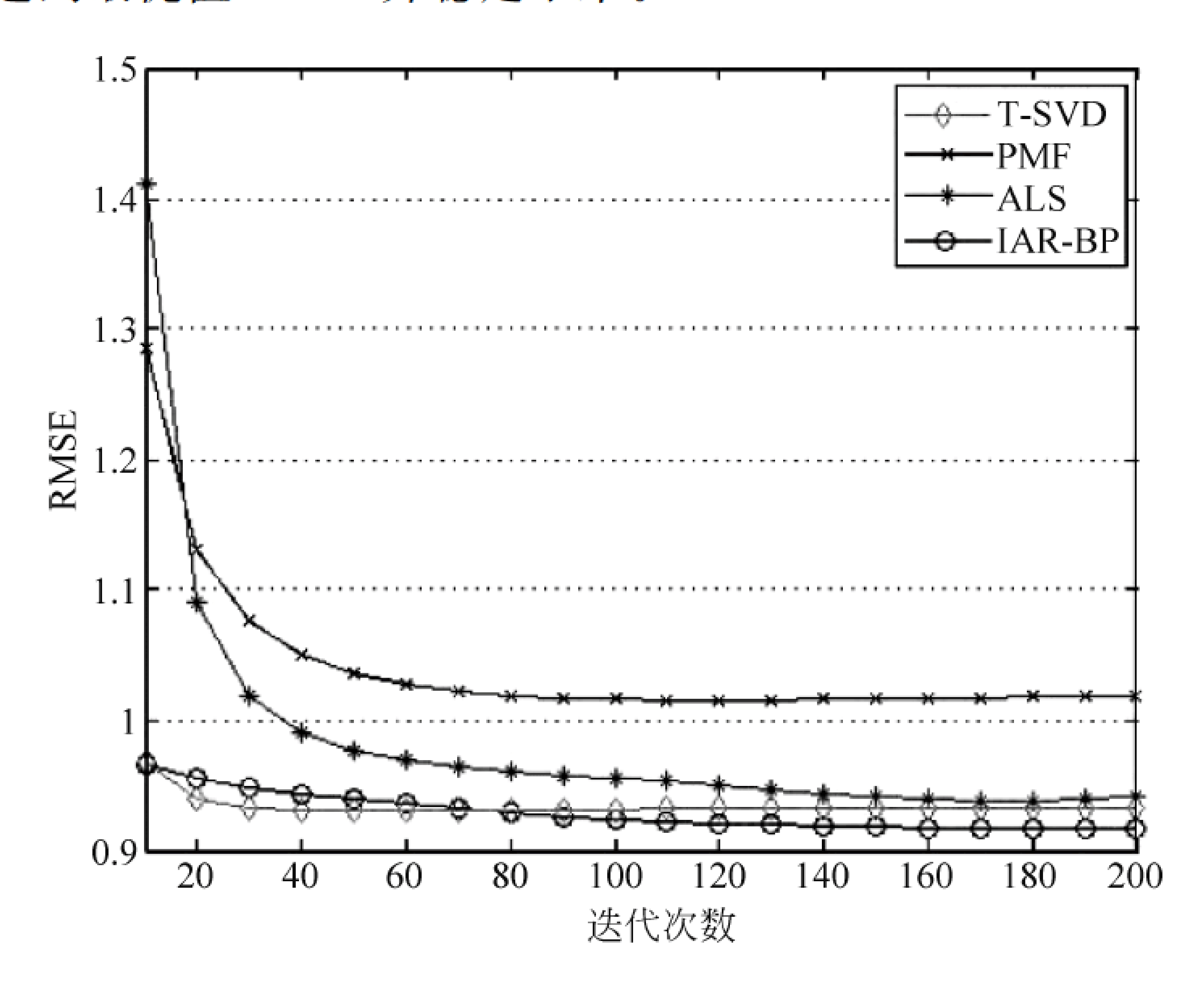


图 7-6 MovieLens 100k 不同算法的 RMSE 值对比

从图 7-6 和图 7-7 中可以看出，在两个数据集上，本章的 IAR-BP 算法都较 PMF 算法在 RMSE 上提升明显，而且收敛快，RBPT 算法在 Ciao 数据集上相对于传统的 PMF 算法精度提升为 13.61%，在 FilmTrust 数据集上精度提升 15.52%，提升效果显著，说明本章提出的的 RBPT 算法具有一定的优越性。

从迭代图，可得到如下结论：

(1) 高效准确地处理大规模数据；

(2) 本章的 IAR-BP 算法比其他单纯利用用户项目评分矩阵信息的方法效果要好。

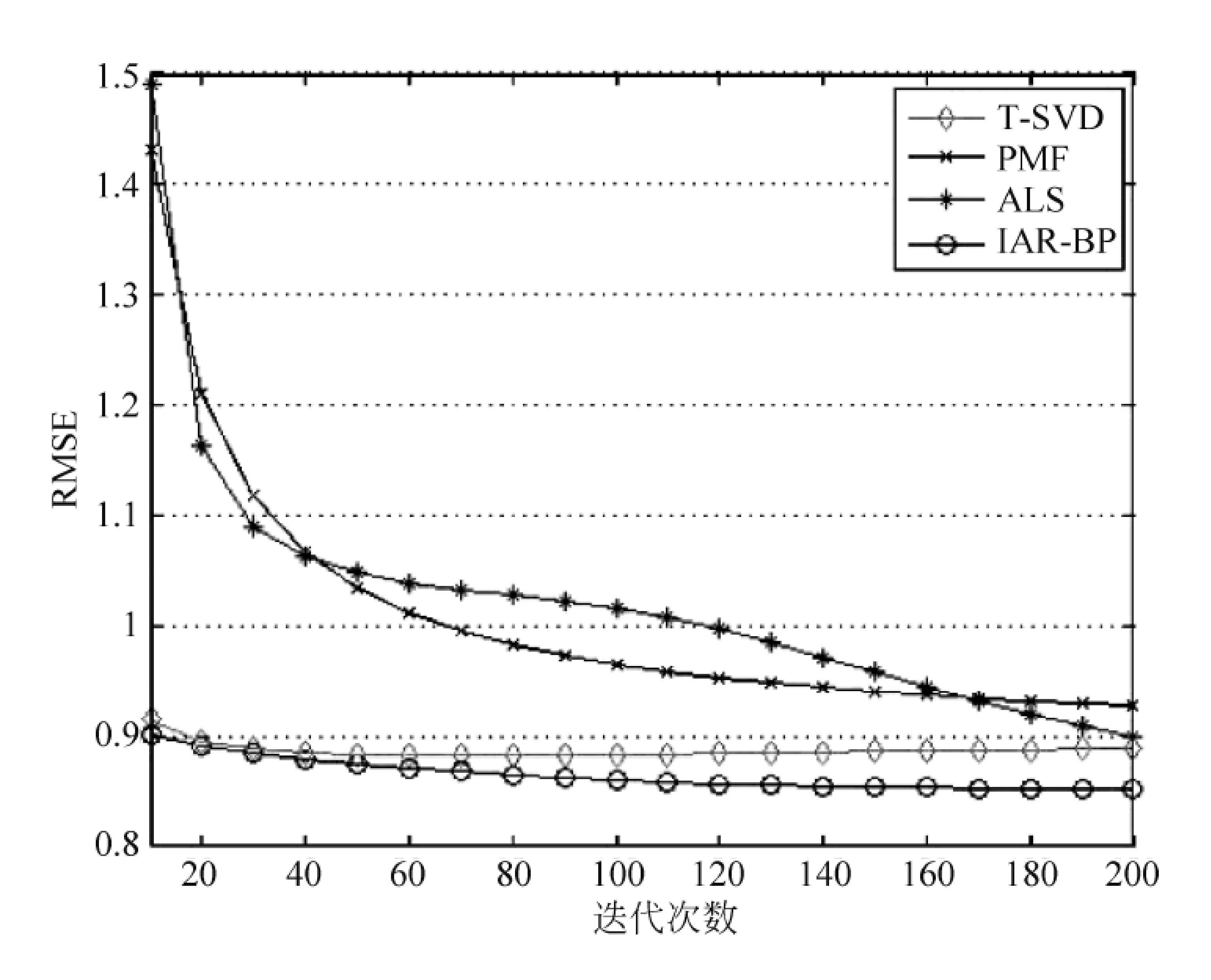


图 7-7　MovieLens 1M 不同算法的 RMSE 值对比

随着迭代次数的增加模型逐步收敛，本章的算法仍然取得较好的结果，表 7-2 所列为不同迭代次数在实验数据集上推荐精度的变化。

表 7-2　不同迭代次数在实验数据集上的推荐精度

数　据　集	迭 代 次 数	T-SVD	PMF	ALS	IAR-BP
MovieLens 100k	10	1.2858	1.4119	0.9683	0.9656
	50	1.0359	0.9769	0.9300	0.9390
	100	1.0160	0.9553	0.9314	0.9241
	150	1.0162	0.9409	0.9327	0.9181
	200	1.0193	0.9407	0.9326	0.9159
MovieLens 1M	10	1.4323	1.4913	0.9150	0.9017
	50	1.0338	1.0481	0.8836	0.8751
	100	0.9647	1.0157	0.8830	0.8602
	150	0.9407	0.9577	0.8861	0.8542
	200	0.9287	0.8990	0.8889	0.8520

表 7-3 所列为不同算法在实验数据集上推荐精度的变化，列出的是不同算法在 Ciao 数据集和 FileTrust 数据上观测到的推荐精度。

表 7-3　不同算法在实验数据集上的推荐精度

数　据　集	算　　法			
	PMF	SocialReg	BiasPMF	IAR-BP
Ciao	1.090	1.0460	0.9573	0.9541
FilmTrust	0.9441	0.8006	0.7950	0.7889

本章小结

本章提出一种基于项目属性改进的概率矩阵分解方法，首先提出一种项目属性相似度的计算方法，然后将相似度作为正则项融入到概率矩阵分解框架中，将项目的潜在因子向量依赖于和其相似的其他项目。最后在不同数据集通过不同的比较，结果表明本章的算法比传统的算法效果要好、训练快、精度高。

参考文献

[1] Ghazanfar M A，Prugel-Bennett A. The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations [J]. Informatics，2013，37(1)：61-92.

[2] Scott D，Dumais S T，Furnas G W，et al. Indexing by latent semantic analysis[J]. Journal of the American Society for Information Science，1990，41(6)：391-407.

[3] 夏小伍，王卫平. 基于信任模型的协同过滤推荐算法[J]. 计算机工程，2011，21(21)：26-28.

[4] Pavlov D，Pennock D M. A Maximum Entropy Approach to Collaborative Filtering in Dynamic，Sparse，High-Dimensional Domains. [C]//In Proceedings of Neural Information Processing Systems，2002：1441-1448.

[5] Ortega F，Hernando A，Bobadilla J，et al. Recommending Items to Group of Users Using Matrix Factorization Based Collaborative Filtering[J]. Information Sciences，2016，345 (C)：313-324.

[6] Zhou X，He J，Huang G，et al. SVD-based incremental approaches for recommender systems[J]. Journal of Computer & System Sciences，2015，81(4)：717-733.

[7] Yang B，Lei Y，Liu J，et al. Social Collaborative Filtering by Trust. [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence，2016，PP(99)：2747-2753.

[8] Zheng X，Luo Y，Sun L，et al. A New Recommender System Using Context Clustering Based on Matrix Factorization Techniques[J]. Chinese Journal of Electronics，2016，25 (2)：334-340.

[9] 赵恒. 基于 LBS 的本地美食推荐系统的研究与实现[D]. 成都：电子科技大学，2015.

[10] Matuszyk P，Vinagre J，Spiliopoulou M，et al. Forgetting methods for incremental matrix factorization in recommender systems[C]//ACM Symposium on Applied Computing. ACM，2015：947-953.

[11] Zhao C，Peng Q，Zhang Z. A Matrix Factorization Algorithm with Hybrid Implicit and Explicit Attributes for Recommender Systems [J]. Journal of Xian Jiaotong University，2016.

[12] Pirasteh P，Hwang D，Jung J J. Exploiting matrix factorization to asymmetric user similarities in recommendation systems[J]. Knowledge-Based Systems，2015，83(1)：51-57.

[13] 何佳知. 基于内容和协同过滤的混合算法在推荐系统中的应用研究[D]. 上海：东华大学，2016.

[14] 刘晓光. 基于遗忘理论和加权二部图的推荐系统研究[D]. 贵阳：贵州大学，2015.

[15] 王立才. 上下文感知推荐系统若干关键技术研究[D]. 北京：北京邮电大学，2012.

[16] 冷亚军，陆青，梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能，2014，27(8)：720-734.

[17] 荣辉桂，火生旭，胡春华，等. 基于用户相似度的协同过滤推荐算法[J]. 通信学报，2014(2)：16-24.

[18] 朱夏，宋爱波，东方，等. 云计算环境下基于协同过滤的个性化推荐机制[J]. 计算机研究与发展，2014，51(10)：2255-2269.

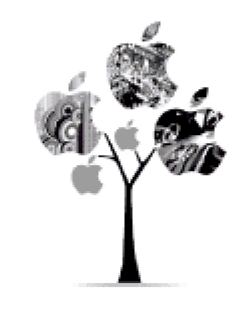
[19] 洪舒怡. 基于矩阵分解的推荐系统模型和算法改进研究[D]. 厦门：厦门大学，2016.

[20] Gomez-Uribe C A，Hunt N. The Netflix Recommender System：Algorithms，Business Value，and Innovation[J]. Acm Transactions on Management Information Systems，2016，6(4)：13.

[21] Rampure V，Tiwari A. A Rough Set Based Feature Selection on KDD CUP 99 Data Set [J]. International Journal of Database Theory & Application，2015，8.

第 8 章

基于交替最小二乘的改进概率矩阵分解算法



本章针对概率矩阵分解对初始值敏感的问题，融合用户项目偏置信息提出一种基于交替最小二乘的改进算法。将分解得到的潜在因子作为交替最小二乘的初始值以提升推荐精度。实验结果表明，本章提出的算法在推荐精度上提升显著。

8.1 引言

概率矩阵分解算法是解决信息过载的有效手段，但在研究中经常面临高维稀疏性带来的推荐精度不高的问题，许多学者提出一些改进算法来缓解。例如，梅忠等人提出一种项目属性约束的概率矩阵分解算法，实验结果验证了改进算法的有效性。然而这类改进算法没有考虑到用户的兴趣差异，不能体现用户间的偏好，导致对于高维稀疏数据推荐效果往往难以得到保证。

结合前人的研究发现，很少有文献针对初始值对概率矩阵分解的影响进行研究，因此本章将结合交替最小二乘算法来训练概率矩阵分解模型，本节的观点基于如下两个既得事实：

(1) 不同用户对不同项目有不同的偏好，例如文章、食物等；

(2) 通过分解用户项目评分矩阵能得到用户和项目的隐式空间矩阵。

8.2 交替最小二乘

理论研究表明，交替最小二乘(Alternating Least Squares，ALS)随着迭代的进行误差会逐步降低直至收敛，ALS 完全不能保证将会收敛至全局最优解，而且在实际应用中，ALS 对初始点选取较为敏感，不恰当的选择会导致数据振荡地收敛到局部最优解。

首先按高斯分布初始化用户和项目的潜在因子向量 U 和 V。

然后固定 V，将损失函数对 V 求偏导，并令导数等于 0，得到新的用户潜在因子向量 U，如式(8-1)所示。

$$U \leftarrow (V^{T}V+\lambda I)^{-1}V^{T}R \tag{8-1}$$

其次固定 U，将损失函数对 U 求偏导，并令导数等于 0，如式(8-2)所示。

$$V \leftarrow (U^{T}U+\lambda I)^{-1}U^{T}R \tag{8-2}$$

式中：λ——正则化系数，需要实验确定。

最后便可利用得到的用户项目潜在因子空间 U 和 V 进行评分预测。

8.3 Baseline 预测

训练推荐算法主要是来训练用户和项目的交互关系，并不是具体的用户项目评分矩阵，如图 8-1 所示是偏置信息。从图 8-1 可以看出 3 个用户(Jim、Lucy 和 Jack)对 3 部电影(A、B 和 C)的评分情况，直观上看用户 Lucy 和 Jim 比较相似，然而其实是 Lucy 和 Jack 比较相似，因为对于电影 A 和 B，Lucy 更偏好电影 B，这和 Jack 是一致的，然而 Jim 却更喜欢电影 A，一般将这些用户的个人评分习惯称为用户偏置信息。



图 8-1　偏置信息

另外，电影本身(导演等因素)和数据集本身(网站等因素)也有偏置信息在内，加入偏置信息的评分预测算法如式(8-3)所示。

$$\hat{R}_{ui} = \mu + bu(u) + bi(i) \tag{8-3}$$

式中：$\hat{R}_{ui}$——用户 u 对项目 i 的预测评分；

μ——数据集的总体偏置信息；

$bu(u)$——用户 u 的偏置信息；

$bi(i)$——项目 i 的偏置信息。

例如，要预测用户 u 对电影 i 的打分，假设电影数据集总体偏置 μ 为 3.4 分，电影 i 的口碑比其他电影高 0.9 分，即 $bi(i)=0.9$；另外，用户 u 是一个悲观的用户，一般偏向于给电影打低分(0.3 分)，即 $bu(u)=-0.3$，那么用户 u 对电影 i 的预测打分为 $3.4+0.9-0.3=4$ 分。加入偏置信息后，目标函数如式(8-4)所示。

$$\underset{bu,bi}{\operatorname{argmin}}\Big(\sum_{u,i}(R_{ui}-\mu-bu(u)-bi(i))^{2}+\lambda_1\sum_{u}bu(u)^{2}+\lambda_2\sum_{i}bi(i)^{2}\Big) \tag{8-4}$$

实际应用中往往根据经验似然采用式(8-5)所示的方法来求解 bi 和 bu 的值。

$$\begin{cases} bi = \dfrac{\sum\limits_{u \in R(i)} (R_{ui} - \mu)}{\lambda_1 + |R(i)|} \\ bu = \dfrac{\sum\limits_{i \in R(u)} (R_{ui} - \mu - bi)}{\lambda_2 + |R(u)|} \end{cases} \tag{8-5}$$

式中：u——某一用户；

i——某一项目；

$R(i)$——评价过项目 i 的用户集合；

$R(u)$——用户 u 评价过的项目集合；

λ_1 和 λ_2——压缩系数，需要实验确定。

8.4 IPMF 算法

8.4.1 算法改进思想

数据的高维稀疏性带来的推荐精度较低的问题是个性化推荐过程中最为显著的问题之一，许多学者提出一系列相关算法来解决，应用到推荐系统中的比较有代表性的矩阵分解方法有 Truncated-SVD 算法以及 ALS 算法。Truncated-SVD 简称 T-SVD 算法，相对传统 SVD 来说只计算用户指定的前 k 维最大奇异值，在不损失推荐精度的前提下总体复杂度更低，具体来说训练得到左奇异矩阵 U、右奇异矩阵 V 和奇异值矩阵 S，然后选择 S 中前 k 维的值奇异值得到 $\sqrt{S(k)}$，初始用户潜在因子矩阵为 $U=U \cdot \sqrt{S(k)}$，初始项目潜在因子矩阵为 $V=V \cdot \sqrt{S(k)}$，ALS 可以认为是 T-SVD 模型的并行化实现，也就是说可以先固定 V(例如，随机初始化 V)，求解 U，然后固定 U，再求解 V，如此交替直至收敛。两种模型在 Netflix 推荐竞赛以及在 KDD Cup 竞赛中都取得了非常优异的成绩，说明这两种模型是当前较为流行的模型，也从侧面证明本章的研究较有价值。

结合前人的研究，本章从用户偏好的角度出发，提出一种融合用户项目偏置信息的概率矩阵分解算法，称为 IPMF(Improved Probabilistic Matrix Factorization)。首先将偏置信息融入到 PMF 中以提升推荐的精度，然后结合最大似然估计把评分预测问题转化为最优化问题，通过 ALS 来求解最优化问题，进而得到 U 和 V，最后结合 U 和 V 的内积进行评分预测。

8.4.2 算法流程

鉴于 PMF 算法并没有考虑到用户项目的偏置信息，同时用户潜在因子矩阵和项目潜在因子矩阵是由均值为 0 的高斯分布随机产生的，这会影响算法的训练速

度和最终的推荐精度，鉴于此往往首先通过 T-SVD 算法对实验数据集进行初步训练，以此作为 IPMF 算法的初始值，同时融入 Bias 信息，如式(8-6)所示是 IPMF 算法最终的目标函数。

$$L=\underset{U,V\mu,bu,bi}{\operatorname{argmin}}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\ (R_{ij}-(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j))^2+\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{\mathrm{Fro}}^2+\frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|_{\mathrm{Fro}}^2+\frac{\lambda_{bu}}{2}\sum_{i=1}^{N}\|bu(i)\|^2+\frac{\lambda_{bi}}{2}\sum_{j=1}^{M}\|bi(j)\|^2 \tag{8-6}$$

式中：μ——数据集的总体偏置；

bu——各个用户的偏置向量；

bi——各个项目的偏置向量；

λ_U、λ_{bu}、λ_V、λ_{bi}——正则化参数。

对式(8-6)求偏导得到如式(8-7)所示的目标函数对于各部分的偏导数计算公式。另外值得注意的一点是，和传统算法不同的是本章的算法需要对数据集的偏置进行更新。

$$\begin{cases}\dfrac{\partial L}{\partial U_i}=\sum_{j=1}^{M}I_{ij}^{R}[(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]V_j+\lambda_U U_i\\ \dfrac{\partial L}{\partial V_j}=\sum_{i=1}^{N}I_{ij}^{R}[(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]U_i+\lambda_V V_j\\ \dfrac{\partial L}{\partial bu(i)}=\sum_{i=1}^{N}I_{ij}^{R}[(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]+\lambda_{bu}bu(i)\\ \dfrac{\partial L}{\partial bi(j)}=\sum_{i=1}^{N}I_{ij}^{R}[(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]+\lambda_{bi}bi(j)\\ \dfrac{\partial L}{\partial \mu}=\sum_{i=1}^{N}I_{ij}^{R}[(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]\end{cases} \tag{8-7}$$

接着按照式(8-8)所示进行参数更新，其中 η 为 SGD 的学习速率。为了降低模型的复杂度，实验中往往人为指定学习速率。

$$\begin{cases}U\leftarrow U-\eta\cdot\dfrac{\partial L}{\partial U}\\ V\leftarrow V-\eta\cdot\dfrac{\partial L}{\partial V}\\ bu\leftarrow bu-\eta\cdot\dfrac{\partial L}{\partial bu}\\ bi\leftarrow bi-\eta\cdot\dfrac{\partial L}{\partial bi}\\ \mu\leftarrow\mu-\eta\cdot\dfrac{\partial L}{\partial \mu}\end{cases} \tag{8-8}$$

如式(8-9)所示是本算法的最终评分预测公式。

$$\hat{R}_{ij} = \mu + bu(i) + bi(j) + U_i^{\mathrm{T}} V_j \tag{8-9}$$

最终的算法流程如算法 8-1 所示。

算法 8-1 IPMF 算法

输入：用户项目评分矩阵 R_{NM}，正则化参数，SGD 迭代次数 T，分解维度 k。
输出：用户和项目潜在因子矩阵 U 和 V，用户偏置 bu，项目偏置 bi 和总体偏置 μ。

```
使用高斯分布初始化模型参数{U=randn(N,k),V=randn(M,k)};
for t =1,2,…,T,do
  随机从 R_NM 选择一条数据;
  根据式(8-5)计算梯度;
    根据式(8-6)更新参数;
end for
ALS 训练;
返回 U 和 V。
算法结束
```

8.4.3 复杂度分析

对 IPMF 进行计算的复杂度为 $o(\rho_R k + MT)$，ρ_R 为 R_{NM} 中已评分元素的个数，各个梯度下降的复杂度为 $o(\rho_R k^2 + MT)$。由于用户在互联网中评分记录服从幂律分布，那么 $M \ll \rho_R$，最终的算法复杂度是 $o(\rho_R k + 5\rho_R k^2)$。

当固定 U 求 V 时，共有 N 个最小二乘子问题，复杂度为 $o(\rho_R k^2 + Nk^2)$，加上固定 V 求 U 时的复杂度，总的复杂度为 $o(\rho_R k + 5\rho_R k^2 + (M+N)k^3)$，可以看出本章提出的算法和数据量 ρ_R 线性相关，适用于大规模数据集。

8.5 实验结果分析

为了不失一般性，一般随机选择 90%作为训练数据，然后预测余下的 10%，通过 5 折交叉验证来确定正则化参数，分解维度选为 $k=5$ 和 $k=10$ 分别进行实验。本章的实验要解决如下几个问题：

(1) Baseline 预测算法的参数设定；

(2) Baseline、T-SVD、PMF 推荐算法作比较；

(3) 正则化参数 λ 和潜在因子维度对对推荐质量的影响。

8.5.1 对比实验设定

1. Baseline 预测

之所以选择 Baseline 预测，是因为其推荐精度高、消耗时间较短。

2．ALS 算法、PMF 算法和 Truncated-SVD 算法

实验中为了降低模型的复杂度，本章人为指定正则化系数 $\lambda_U=\lambda_V=0.01$，梯度下降的学习速率 $\eta=0.03$。

8.5.2　实验分析

1．BaseLine 预测

如图 8-2 所示是在 MovieLens 数据集上做的 Baseline 预测。经实验发现，当 $\lambda_1=2$、$\lambda_2=5$ 时 RMSE 达到最优，最小值为 0.9559。

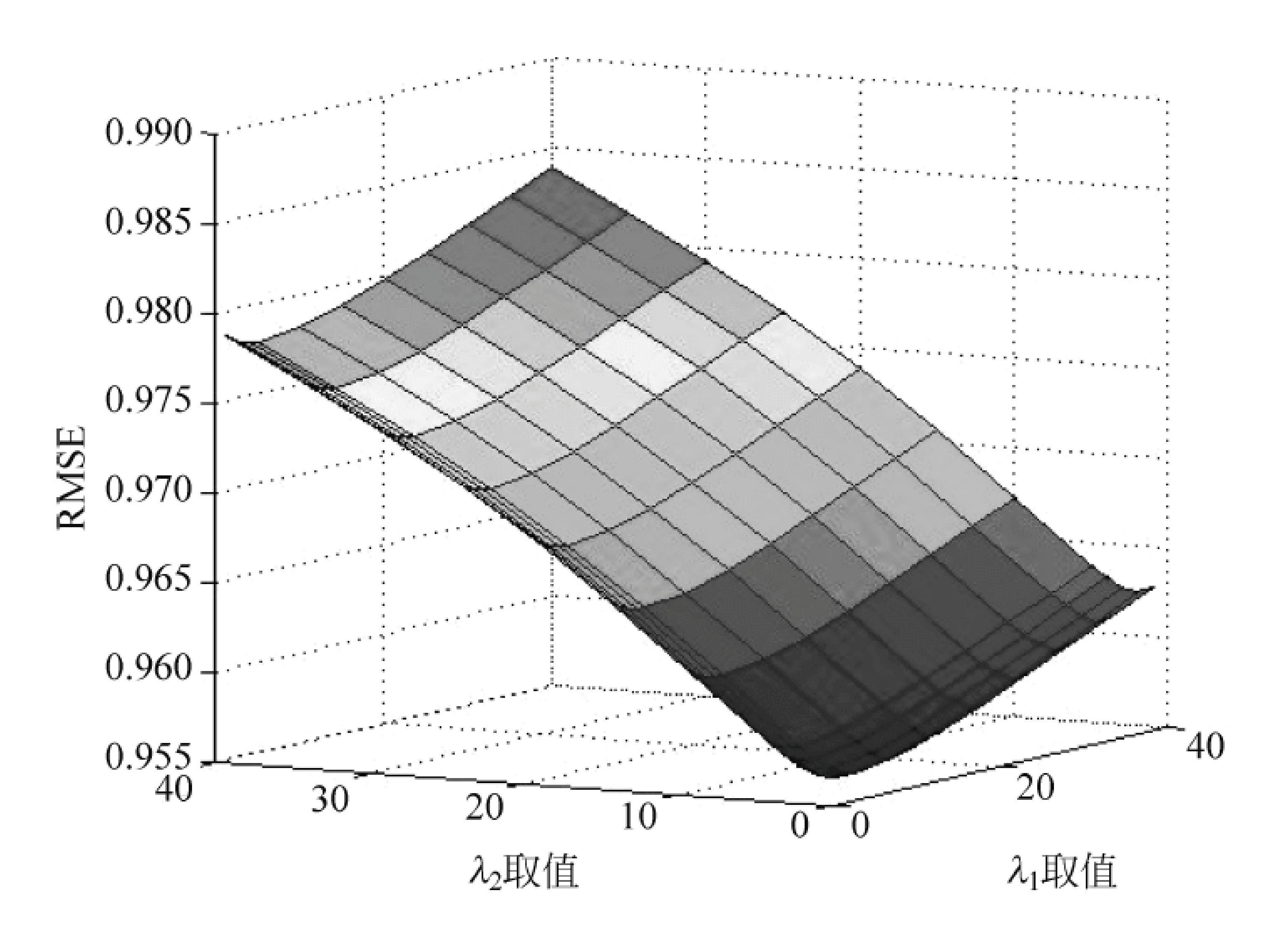


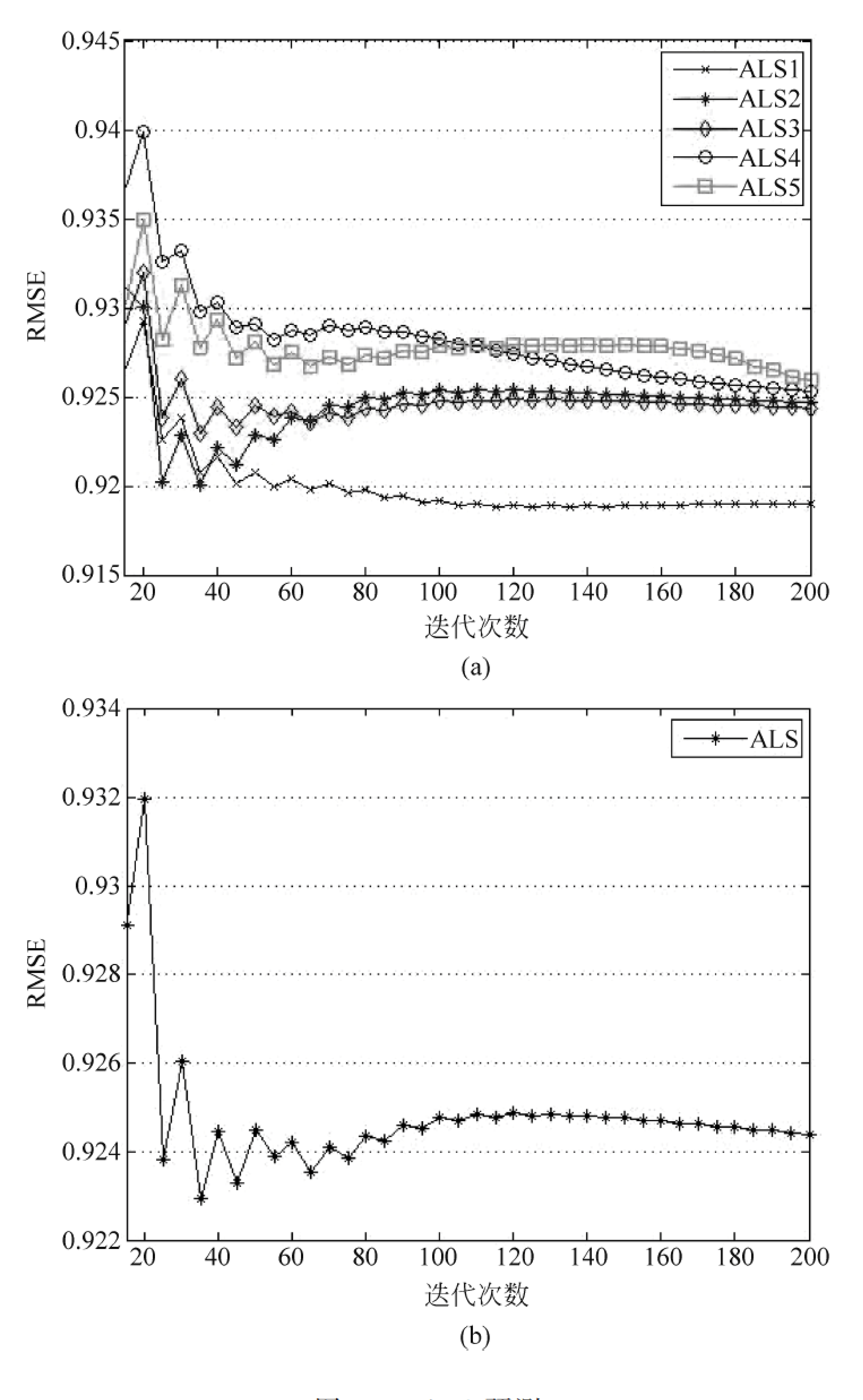
图 8-2　Baseline 预测

2．ALS 算法

如图 8-3 是 ALS 预测，该预测是标准 ALS 算法在 MovieLens 100k 数据集上的实验结果，图 8-3(a)是随机进行 5 次 ALS 实验，图 8-3(b)是其中某一次的 ALS。从图中可以看出，由于 ALS 算法初始值的随意选择，导致 ALS 每次的结果都不一样，而且初始迭代比较振荡，说明标准的 ALS 算法对初始值较为敏感，不仅影响推荐精度，也影响训练速度。

3．ALS-PMF 对比

如图 8-4 是在不同潜在因子维度 k 下 PMF 与 ALS 的算法对比图，图 8-4(a)是 $k=5$ 的情况，图 8-4(b)是 $k=10$ 的情况。从图中可以看出，不同潜在因子维度 k 下，标准 ALS 和 PMF 算法的迭代曲线基本不变，但是最终的推荐精度不一样，说明在实际应用中潜在因子维度的选择十分重要。另外值得注意的是，并不是 k 越大推荐精度越高，这是因为矩阵分解算法假设原始评分矩阵是低秩的，过大的 k 可能会引入噪声，进而降低推荐精度。



(a)

(b)

图 8-3 ALS 预测

4. 正则化参数

不仅潜在因子维度影响推荐精度，正则化参数也是影响推荐精度的重要因素，如图 8-5 所示为不同 k 值 RMSE 随着 λ 的变化曲线，图 8-5(a)是 $k=5$ 时正则化参数 λ 对推荐质量的影响，图 8-5(b)是 $k=10$ 时 λ 对推荐质量的影响。从图中可以看出，当 λ 小于一定值时，RMSE 值基本处于下降的状态；随着 λ 再次增加，RMSE 又开始处于上升的态势，最终在 $k=5,\lambda=0.4$ 时，RMSE 达到最优；在 $k=10,\lambda=0.6$ 时，RMSE 达到最优。

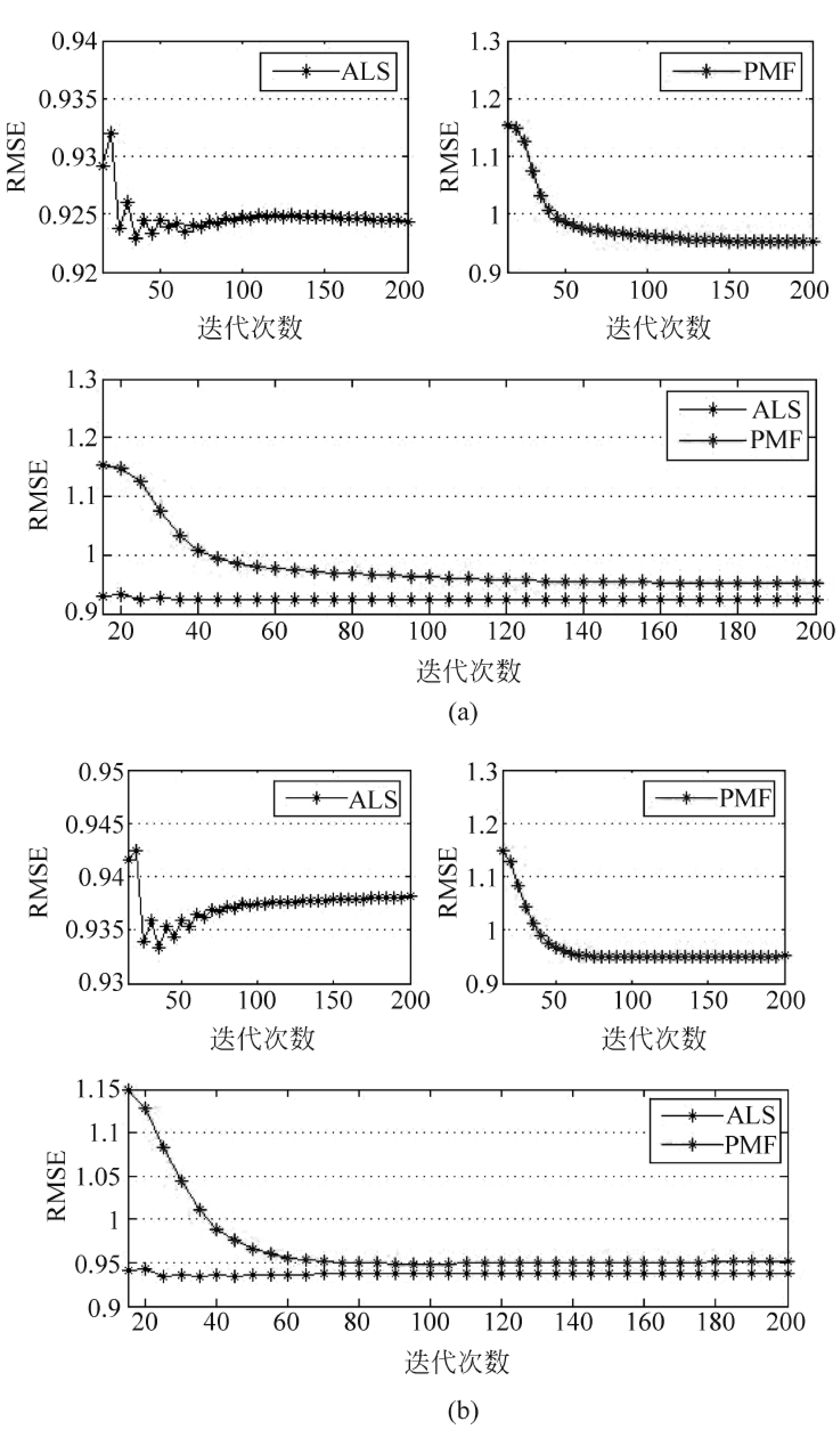


图 8-4　ALS 与 PMF 在不同 k 值的对比

5．实验对比

为了更直观地凸显本章算法的优点，绘制如图 8-6 所示不同算法的推荐质量随迭代次数的变化情况，所含的算法包括 ALS、T-SVD、PMF、IPMF 随着迭代次数在不同数据及上的变化曲线。从图 8-6 中可以看出，本章提出的算法无论 k 为多少，在 RMSE 上的推荐精度都是最高的；同时从图中可以看出本章算法结果较为稳定，从侧面反映出本章的算法具有一定的鲁棒性。

表 8-1 为不同 k 对应的 RMSE 值。本章给出不同算法在不同 k 值下最终的 RMSE，$k=5$ 时，本章算法相对于标准的 PMF 提高 3.41%，$k=10$ 时提高 2.23%。

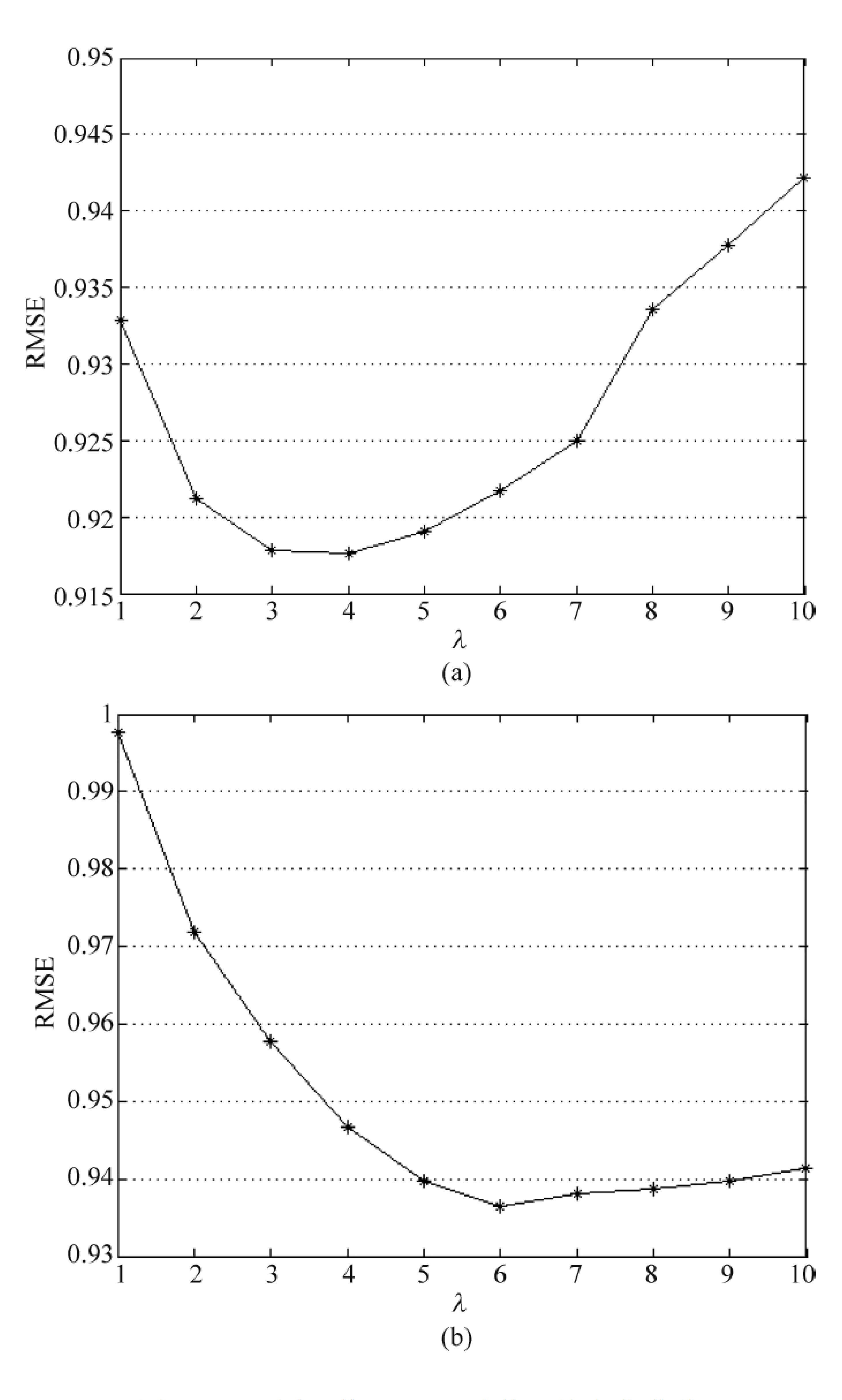


图 8-5 不同 k 值 RMSE 随着 λ 的变化曲线

表 8-1 不同 k 值对应的 RMSE 值

k	RMSE				
	Baseline	T-SVD	PMF	ALS	IPMF
$k=5$	0.9553	0.9535	0.9517	0.9329	0.9176
$k=10$	0.9553	1.0154	0.9488	0.9396	0.9265

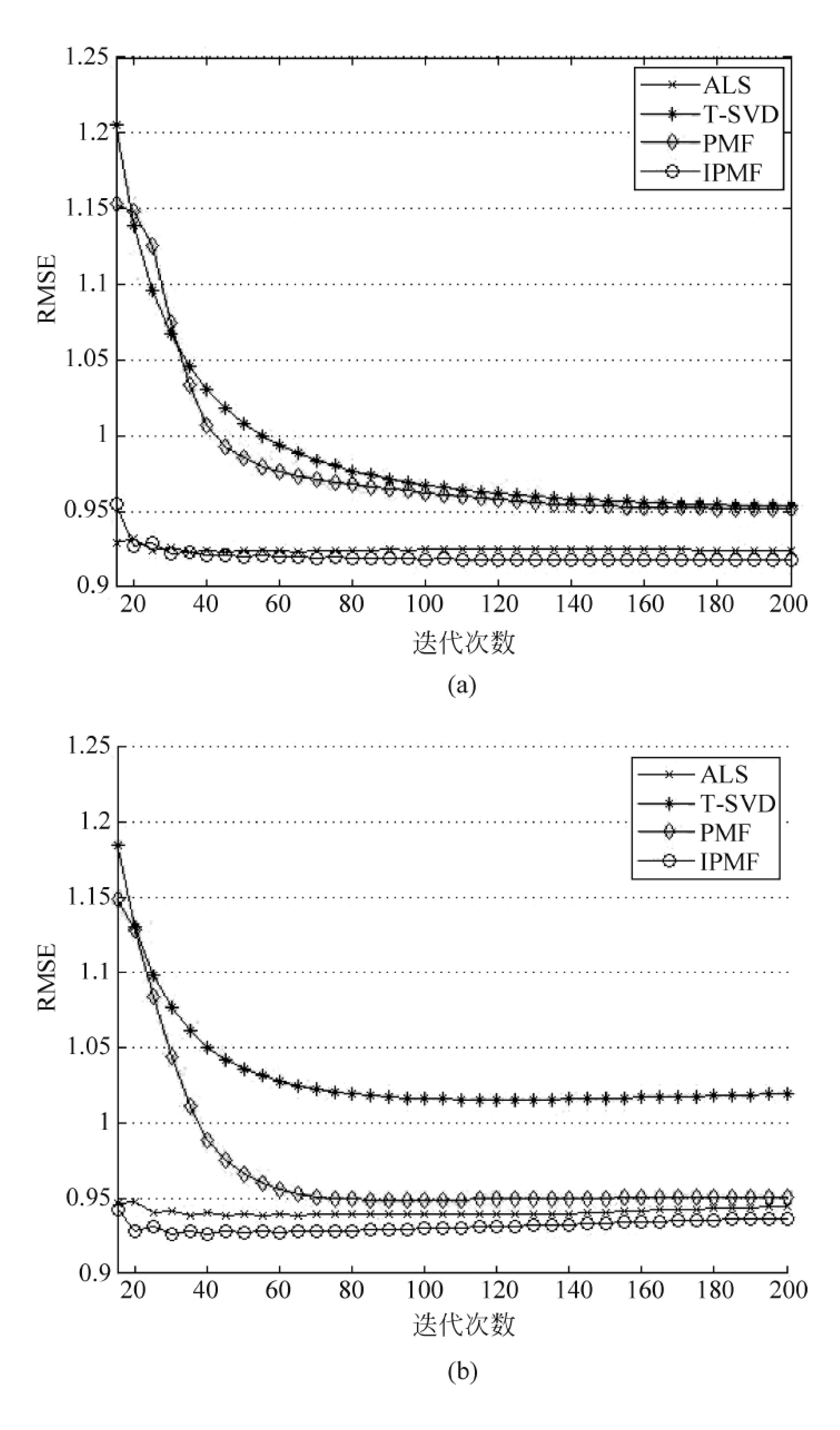


图 8-6　不同算法的推荐质量随迭代次数的变化情况

本章小结

在根据传统 PMF 和其他矩阵分解方法进行研究的基础上，本章针对优化的方法展开研究，提出一种 ALS 和用户项目偏置信息优化的概率矩阵分解算法，最后通过实验验证提出算法的有效性。

参考文献

[1] 梅忠，肖如良，张桂刚. 基于受约束偏置的概率矩阵分解算法[J]. 计算机系统应用，2016，25(5)：113-117.

[2] 陆园丽. 基于非负矩阵分解的鲁棒推荐算法研究[D]. 秦皇岛：燕山大学，2015.

[3] Ortega F，Hernando A，Bobadilla J，et al. Recommending items to group of users using Matrix Factorization based Collaborative Filtering[J]. Information Sciences，2016，345(C)：313-324.

[4] Zhao X，Niu Z，Chen W，et al. A hybrid approach of topic model and matrix factorization based on two-step recommendation framework[J]. Journal of Intelligent Information Systems，2015，44(3)：335-353.

[5] Zhao Y，Li S，Hou J. Link Quality Prediction via a Neighborhood-Based Nonnegative Matrix Factorization Model for Wireless Sensor Networks[J]. International Journal of Distributed Sensor Networks，2015，2015(1)：1-8.

[6] Solovyev S A，Tordeux S. An efficient truncated SVD of large matrices based on the low-rank approximation for inverse geophysical problems[J]. Université De Pau Et Des Pays De Ladour，2015：592-609.

[7] Mori K，Nguyen T，Harada T，et al. An Improvement of Matrix Factorization with Bound Constraints for Recommender Systems[C]//Iiai International Congress on Advanced Applied Informatics. 2016：103-106.

[8] Aleksandrova M，Brun A，Boyer A，et al. Identifying representative users in matrix factorization-based recommender systems：application to solving the content-less new item cold-start problem[J]. Journal of Intelligent Information Systems，2016：1-33.

[9] He X，Zhang H，Kan M Y，et al. Fast Matrix Factorization for Online Recommendation with Implicit Feedback[C]//International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM，2016：549-558.

[10] Hanhua Chen，Hai Jin，Xiaolong Cui. 微博系统中一种混合关注对象推荐方法[J]. Science China Information Sciences，2017，60(1)：012102.

[11] Ma T，Zhou J，Tang M，et al. Social Network and Tag Sources Based Augmenting Collaborative Recommender System[J]. Ieice Transactions on Information & Systems，2015，E98.D(4)：902-910.

[12] Hong M，Jung J J. MyMovieHistory：Social Recommender System by Discovering Social Affinities Among Users[J]. Cybernetics & Systems，2016，47(1-2)：88-110.

[13] Salakhutdinov R，Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]//International Conference on Machine Learning. ACM，2008：880-887.

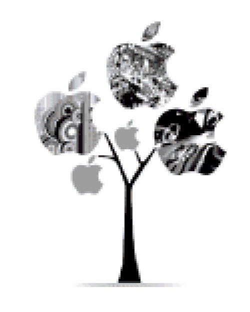
[14] Rendle S. Factorization Machines with libFM[J]. Acm Transactions on Intelligent Systems & Technology，2012，3(3)：57.

[15] Guo G，Zhang J，Sun Z，et al. Librec：A java library for recommender systems[C]//Posters，Demos，Late-breaking Results and Workshop Proceedings of the 23rd International Conference on User Modeling，Adaptation and Personalization. 2015：955-963.

[16] Fernández-Tobías I, Braunhofer M, Elahi M, et al. Alleviating the new user problem in collaborative filtering by exploiting personality information[J]. User Modeling and User-Adapted Interaction, 2016, 26(2): 221-255.
[17] 王立才,孟祥武,张玉洁. 上下文感知推荐系统[J]. 软件学报, 2012, 23(1): 1-20.
[18] He R, Mcauley J. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering[C]//International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2016: 507-517.
[19] Silva E Q D, Camilo-Junior C G, Pascoal L M L, et al. An evolutionary approach for combining results ofrecommender systems techniques based on collaborative filtering[J]. Expert Systems with Applications, 2016, 53: 204-218.
[20] Zhang J, Lin Y, Lin M, et al. An effective collaborative filtering algorithm based on user preference clustering[J]. Applied Intelligence, 2016, 45(2): 1-11.
[21] Yang B, Lei Y, Liu J, et al. Social Collaborative Filtering by Trust. [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, PP(99): 2747-2753.
[22] Zheng X, Luo Y, Sun L, et al. A New Recommender System Using Context Clustering Based on Matrix Factorization Techniques[J]. Chinese Journal of Electronics, 2016, 25 (2): 334-340.
[23] Kushwaha N, Sun X, Vyas O P, et al. SemPMF: Semantic Inclusion by Probabilistic Matrix Factorization for Recommender System[J]. 2016, 27(2): 294-310.
[24] Qiu Y, Lin C J, Juan Y C, et al. recosystem: Recommender System using Matrix Factorization[J]. 2015, 45(3): 39-47.

第 9 章

基于社交网络的改进概率矩阵分解算法研究



针对社交网络中数据稀疏性带来的推荐精度不高的问题，本章提出利用用户—项目评分矩阵和用户的社交网络信息来解决数据稀疏性带来的推荐精度不高的问题。首先借鉴用户普遍的认知心理，将信任作为实体决策选择时的一个主观概念，结合客观存在的评分，共同评价用户之间的偏好关系。然后将用户偏置信息和用户间的信任关系融入传统的概率矩阵分解中，通过随机梯度下降来获取最优解，从而获得对原始用户—项目评分矩阵中缺失的评分值的预测。实验结果表明本章提出的算法较传统的模型在推荐精度上有较大提高，尤其是相对于 PMF(Probabilistic Matrix Factorization)算法在精度方面提高幅度为 13%～15%。

9.1 引言

随着 Web 3.0 的社交网络平台的兴起以及电子商务越来越受到人们的关注，网络中的信息逐渐增多，信息总量超过了个人或系统所能接受、处理或有效利用的范围，大量无关的冗余数据信息严重干扰了受众对相关有用信息的准确选择，用户寻找自己需要的信息越来越困难，这就是所谓的“信息过载”。信息过载是信息时代信息过于丰富的负面影响之一，在信息过载背景下，推荐系统应运而生。推荐系统是通过分析用户或者项目的偏好信息，找到相似用户群体或者相似项目群体，然后根据相似用户群体或者项目群体来为目标用户提供个性化的推荐，包括视频、书籍、音乐和新闻等。由于巨大的潜在商业价值，推荐系统得到飞速发展，成功的推荐系统如商品推荐的亚马逊和电影推荐的 Netflix 公司。

尽管推荐系统在工业界取得巨大成功，然而在实际应用中也存在很多问题，其中最为典型的是因为评分矩阵的稀疏性而造成的推荐精度不高问题。由于网络中的数据越来越多，只有很少数的用户会对同一个项目进行评分，有很多项目根本没有评分，因此数据稀疏性问题越来越严重，推荐系统的推荐精度也在逐渐下降。

为了提高推荐的精度，国内外学者进行了广泛的研究，矩阵分解模型因为具有较好的推荐效果和扩展性受到了越来越多的关注。国外比较有代表性的是 Ruslan Salakhutdinov 等人在文献[1]提出的概率矩阵分解算法并在同年在文献[2]提出贝叶斯概率矩阵分解算法；Koren 在文献[3]中对隐语义模型进行了描述，并在此基础上结合用户的显式反馈和隐式反馈提出了一种称为 BiasedMF 的推荐算法；Steffen Rendle 在文献[4]中利用其所提出 libFM 实现因子分解机的相关算法。国内比较有代表性的有东北大学郭贵冰副教授所领导的推荐系统团队在文献[5]设计开发的 libRec 推荐系统库，中国台湾师范大学林智仁教授所领导的机器学习小组研发的业内著名的 libMF 矩阵分解库并在文献[6]中予以描述。

不过单纯依靠用户项目评分信息并不能显著提高推荐精度，同时传统推荐算法往往有很严重的“马太效应”，也就是说推荐的商品往往都是热门的商品，这样造成热门的商品更加热门，而处在“长尾分布”上的商品得不到重视，为此人们不得不寻找其他资源，其中信任网络就是一种可以利用的资源。例如，文献[7]提出一种基于用户社交间社交关系的推荐系统，文献[8]提出一种改进的信任传播算法并应用到基于图的聚类算法中，实验结果表明加入信任不仅推荐结果精准而且往往会有意想不到的效果。实际生活中也是如此，例如，当一个人选择去餐馆时，很可能会根据自己的口味然后结合好友推荐的意见选择菜品。

本章主要针对 PMF(Probabilistic Matrix Factorization)算法进行研究，PMF 把用户项目评分矩阵作为唯一的信息源，忽略了用户之间的社会关系，导致推荐结果可能偏离用户的需求；同时，由于算法假设用户和用户之间、项目和项目之间独立同分布，没有考虑到用户之间的社交关系以及用户和项目本身的差异，不能显著提高推荐质量，尤其对于高维稀疏数据推荐效果往往难以得到保证。

本章的观点基于如下假设：

(1) 不同用户对同一项目有不同的偏好，就电影来说，并不是所有人都喜欢恐怖片；同理，用户对不同项目也有不同的偏好。

(2) 用户项目评分矩阵包含冗余信息，用户和项目的隐式特征能够通过分解用户—项目评分矩阵得到。

(3) 用户会很容易受其所信任的朋友影响，从而偏好于朋友的推荐，即如果用户 u_1 信任用户 u_2，那么这两个用户的消费记录也会很相似。

基于以上假设，本章从信任的角度出发，提出一种称为 RBPT(Recommendation with Bias Probabilistic Matrix Factorization and Trust relations)的算法，利用用户项目评分矩阵和用户的社交网络信息来解决数据稀疏性带来的推荐精度不高的问题。实验结果表明，本章的算法能够有效缓解由数据的高维稀疏性带来的推荐精度不高的问题。

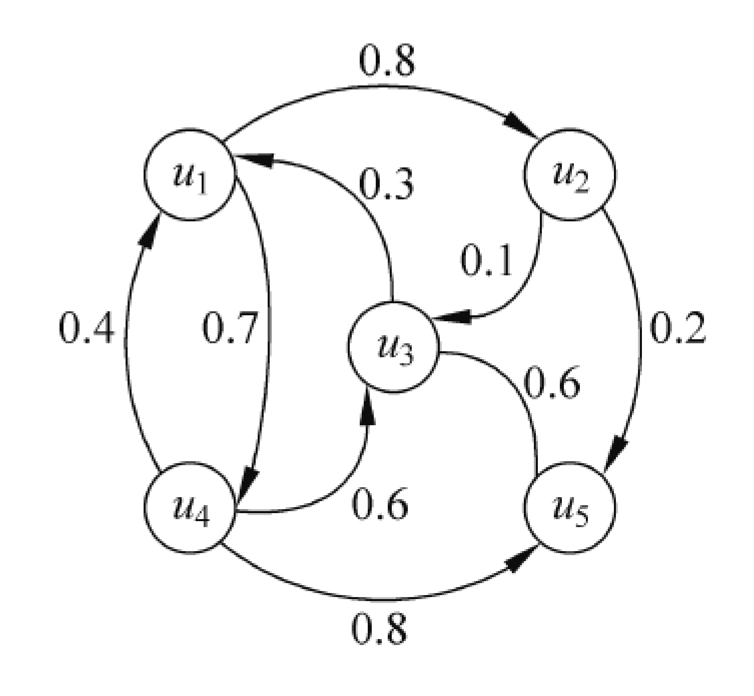
9.2 相关工作

9.2.1 推荐系统的形式化

一个典型的推荐系统，包括一个含有 N 个用户的用户集合 $U=\{u_1,u_2,u_3,\cdots,u_N\}$ 和 M 个项目的集合 $I=\{i_1,i_2,i_3,\cdots,i_M\}$，每个用户 $u\in U$ 评价了 I 中的一部分项目，评价过的项目用 $I_u\subseteq I$ 表示，用户的打分记录往往表示成 R_{NM}，每一个实体 R_{ui} 表示用户 u 对项目 i 的评分，通常 R_{ui} 取 1～5 的整数值，数据越大表示用户对该项目越满意。实际中用户—项目评分矩阵 R_{NM} 非常稀疏，因为用户往往只是评价过一部分项目，例如 Epinions 数据集和 Netflix 数据集中已有的评分数目所占比例都不足 1%。正是因为稀疏性，传统的推荐算法的质量才会特别差。

在真实世界中，以商品购买为例，用户的购买意图受两方面的影响，即用户本身的需要和用户的朋友的推荐程度。如图 9-1 所示为基于社交网络的推荐机制示例，图 9-1(a)所示是用户的信任网络图，该图是一个有向图，包含 5 个节点(用户数)，9 条边(用户信任关系数)，每个节点代表一个用户，节点 i 到节点 j 的边表示用户 u_i 信任用户 u_j，边的权重表示信任程度的大小。注意用户间的信任关系是非对称的，例如用户 u_1 信任 u_2，但是 u_2 就不信任 u_1。

图 9-1(b)是对应的用户—项目评分矩阵，矩阵中已有的值表示用户对项目的评分，缺失部分是需要预测的。以看电影为例，假设用户 u_1 想看电影 v_4，但是该用户对该电影一无所知，那么该用户就会求助于其所信任的朋友 u_2 和 u_4，u_2 对该电影的评分是 3 分，u_4 是 5 分，那么该电影很可能会吸引到用户 u_1，也就是 u_1 对 v_4 的评分也可能很高。系统的目标就是利用评分矩阵和信任关系精准有效地预测用户对未评过分项目的评分。



(a) 用户间社交网络

u	i				
	i_1	i_2	i_3	i_4	i_5
u_1		5	2		3
u_2	4			3	
u_3			2	4	
u_4	5			3	1
u_5		5	5		

(b) 用户项目评分矩阵

图 9-1　基于社交网络的推荐机制示例

9.2.2　矩阵分解与推荐系统

近年来矩阵分解技术由于其良好的可扩展性和较高的推荐精度吸引了国内外学者的广泛关注，尤其是经过 Netflix 推荐竞赛后。矩阵分解技术假设用户的偏好和项目的特征能通过一些潜在因子来描述，对于原始评分矩阵，找到一个与原始评分矩阵 R 相符合的低秩近似矩阵 R'，使得原始矩阵 R 中的实际评分与 R' 中的预测评分之间的距离平方和最小。对于 $N\times M$ 的用户—项目评分矩阵 R，矩阵分解的目标是找到合适的值 k，结合现有的数据利用 $R'=U^{\mathrm{T}}V$ 来近似原有的矩阵，其中 $U\in R^{kN}$，$V\in R^{kM}$，再利用计算出的用户和项目的特征向量就可以对原始用户—项目评分矩阵中缺失的评分值进行进一步预测。比较有代表性的有概率矩阵分解、贝叶斯概率矩阵分解以及由中国台湾林智仁教授所提出的快速并行矩阵分解。

基于矩阵分解的推荐算法是一种学习型算法，通过优化预先设定的目标函数从而得到全局最优解，而且由于潜在因子的数量 $k<<\min(m,n)$，算法离线计算的空间复杂度低，这在当今大数据的环境下具有很强的实用价值；同时，由于该算法有一个全局的目标函数使得算法的预测准确率高。优化目标函数往往通过随机梯度下降(Stochastic Gradient Descent，SGD)来实现，它是基于这样的事实：如果函数在某点处可微且有定义，那么在该点处沿着梯度相反的方向函数值增加得最快；如果要求出目标函数的极小值，那么在迭代的每一步可以沿着当前点的负梯度方向搜索下一个点，使得每次迭代都能够使目标函数值逐步减小，直至逼近目标函数的局部极小值点。

9.3　概率矩阵分解

概率矩阵分解模型是矩阵分解模型中应用非常成功的一个推荐模型，由 H. Shan 和 A. Banerjee 于 2007 年提出，该模型在 Netflix 推荐竞赛以及 KDD Cup 竞赛中都取得了非常优异的成绩。

概率矩阵分解的基本思想是在矩阵分解的基础上引入概率的思想，首先假设用户对项目的真实评分与预测评分的误差评分服从均值为 0 的高斯分布，如式(9-1)所示。

$$R_{ij}\leftarrow U_i^{\mathrm{T}}V_j\Rightarrow p(R_{ij}-U_i^{\mathrm{T}}V_j\,|\,0,\sigma^2)$$
$$\Leftrightarrow p(R_{ij}\,|\,U_i^{\mathrm{T}}V_j,\sigma^2) \tag{9-1}$$

其次，假设用户和商品的特征向量矩阵都符合均值为 0 的高斯分布，如式(9-2)所示。

$$\begin{cases} p(U\,|\,\sigma_U^2)=\prod\limits_{i=1}^{N}[N(U_i\,|\,0,\sigma_U^2)] \\ p(V\,|\,\sigma_V^2)=\prod\limits_{j=1}^{M}[N(V_j\,|\,0,\sigma_V^2)] \end{cases} \tag{9-2}$$

另外，假设用户之间的评分独立同分布，服从球形高斯先验分布，那么评分的

条件概率分布如式(9-3)所示。

$$p(R|U,V,\sigma^2)=\prod_{i=1}^{N}\prod_{j=1}^{M}[N(R_{ij}|U_i^{\mathrm{T}}V_j,\sigma^2)]^{I_{ij}} \tag{9-3}$$

式中：$N(R_{ij}|U_i^{\mathrm{T}}V_j,\sigma^2)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(R_{ij}-U_i^{\mathrm{T}}V_j)^2}{2\sigma^2}\right)$。

利用贝叶斯推导，可得用户和物品的隐式特征的后验概率，如式(9-4)所示。

$$p(U,V \mid R,\sigma^2,\sigma_V^2,\sigma_U^2)\propto p(R \mid U,V,\sigma^2)\times p(U \mid \sigma_U^2)\times p(V \mid \sigma_V^2) \tag{9-4}$$

对上述预测公式取对数，如式(9-5)所示。

$$\begin{aligned}\ln p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2)=&-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\ (R_{ij}-U_i^{\mathrm{T}}V_j)^2-\\&\frac{1}{2\sigma_U^2}\sum_{i=1}^{N}U_i^{\mathrm{T}}U_i-\frac{1}{2\sigma_V^2}\sum_{j=1}^{M}V_j^{\mathrm{T}}V_j-\\&\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\right)\ln\sigma^2+ND\ln\sigma_U^2+MD\ln\sigma_V^2\right)+C\end{aligned} \tag{9-5}$$

式中，C——不依赖于参数的常数。

最大化 U 和 V 的后验概率等于最小化式(9-6)。

$$\begin{aligned}L(U,V)=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\ (R_{ij}-U_i^{\mathrm{T}}V_j)^2+\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{\mathrm{Fro}}^2+\\&\frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|_{\mathrm{Fro}}^2\end{aligned} \tag{9-6}$$

式中：$p(A|B)$——事件 B 发生的情况下事件 A 发生的条件概率；

λ_U 和 λ_V——正则化系数，防止过拟合；

$N(x|\mu,\sigma^2)$——期望为 μ；方差为 σ 的高斯分布；

I——指示函数，如果用户 i 选择了商品 j，$I_{ij}=1$，否则为 0。

为了不失一般性，往往用 $f(x)=(x-1)/(R_{\max}-1)$ 把实际评分映射到 $(0,1]$，用 $g(x)=1/(1+\exp(x))$ 把预测评分映射到 $(0,1]$，$R_{\max}$ 为数据集中的最大评分值。

最终目标函数如式(9-7)所示。

$$\begin{aligned}L(U,V)=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\ (R_{ij}-g(U_i^{\mathrm{T}}V_j))^2+\\&\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{\mathrm{Fro}}^2+\frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|_{\mathrm{Fro}}^2\end{aligned} \tag{9-7}$$

式中：Fro——矩阵的 F 范数。

9.4 主要研究内容

9.4.1 基于社交网络的改进概率矩阵分解

如何借助其他信息改善推荐系统的推荐质量成为推荐领域广泛关注的热点问

题，下面将首先对用户项目的偏置信息进行说明，然后阐述如何将用户的社交网络信息融入矩阵分解，并描述本章所提出的 RBPT 算法，最后对本章的对比实验进行说明。

1. 用户项目偏置信息

观测到的评分数据有一些是和用户或物品无关的因素产生的效果，即一部分因素是和用户对物品的喜好无关而只取决于用户或物品本身特性。例如，乐观型用户的评分行为普遍偏高，而悲观型用户的评分记录普遍偏低，也就是说即使这两类用户对同一项目的评分相同，但是对该物品的喜好程度却并不一样。同理，对同一件(类、种)物品来说，以电影为例，受大众欢迎的电影得到的评分普遍偏高，而一些烂片的评分普遍偏低，这些因素都是独立于用户或产品的因素，而和用户对物品的喜好无关。本章将这些独立于用户或物品的因素称为偏置(Bias)信息，加入偏置信息的评分预测算法如式(9-8)所示。

$$\hat{R}_{ui} = \mu + bu(u) + bi(i) \tag{9-8}$$

式中：$\hat{R}_{ui}$——用户 i 对项目 j 的预测评分；

μ——数据集的总体偏置信息；

$bu(i)$——用户 i 的偏置信息；

$bi(j)$——项目 j 的偏置信息。

例如，要预测用户 user1 对电影 movie1 的打分，假设电影数据集总体偏置 μ 为 3.4 分，电影 movie1 的口碑比其他电影高 0.9 分，即 $bi(\text{movie1}) = 0.9$；另一方面，用户 user1 是一个悲观的用户，一般偏向于给电影打低分(0.3 分)，即 $bu(\text{user1}) = -0.3$，那么用户 user1 对电影 movie1 的预测打分为 $3.4 + 0.9 - 0.3 = 4$ 分。

2. 社交网络正则化

如果 u_i 信任 u_d，那么这两个用户的用户潜在因子空间也要很相似，相似程度取决于 u_i 对 u_d 的信任程度，通过最小化用户 u_i 和 u_d 的欧氏距离，如式(9-9)所示。

$$\min_U \frac{1}{2}\sum_{i=1}^{N}\sum_{d\in \text{Trust}(i)} T_{id} \ \| u_i - u_d \|_F^2 \tag{9-9}$$

式中，Trust(i)——用户 u_i 信任的用户集合，信任的计算如式(9-10)所示。

$$T_{id} = \sqrt{\frac{\Delta'^{-}(u_d)}{\Delta^{+}(u_i) + \Delta^{-}(u_d)}} \tag{9-10}$$

式中：$\Delta^{+}(u_i)$——用户 u_i 在信任网络中的出度，也就是用户 u_i 信任的用户个数；

$\Delta^{-}(u_d)$——用户 u_d 在信任网络中的入度，也就是用户 u_d 被信任的用户信任的次数。

式(9-10)出于现实中这样的考虑：一个用户被信任的次数越多，那么越值得被信任；一个用户信任的人越多，那么其对陌生人也就更倾向于信任，但是信任程度会较低，开根号是为了降低信任关系对模型的影响。

3. RBPT 算法

结合偏置信息和社交网络正则化，本章提出一种称为 RBPT 的算法模型，在传统 PMF 算法的基础上，结合用户项目的偏置信息来提高推荐精度，同时借鉴用户普遍的认知心理将信任作为实体决策时的一个主观考量，通过在 PMF 目标函数最后加入如式(9-9)所示的正则项来进一步提高推荐精度，本算法的最终目标如式(9-11)所示。

$$L(R,T,U,V)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}^{R}[R_{ij}-g(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V)]^2+\frac{\beta}{2}\sum_{i=1}^{N}\sum_{d\in \mathrm{Trust}(i)}T_{id}\|u_i-u_d\|_F^2+\frac{\lambda_U}{2}\|U\|_F^2+\frac{\lambda_V}{2}\|V\|_F^2+\frac{\lambda_{bu}}{2}\sum_{i=1}^{N}\|bu(i)\|_+^2\frac{\lambda_{bi}}{2}\sum_{i=1}^{M}\|bi(j)\|^2 \tag{9-11}$$

式中：Trust(i)——用户 u_i 信任的用户集合；

λ_{bu} 和 λ_{bi}——用户项目偏置信息的正则化系数，防止过拟合。

最终的评分预测算法如式(9-12)所示。

$$\hat{R}_{ui}=\mu+bu(u)+bi(i)+U_i^{\mathrm{T}}V_j \tag{9-12}$$

式中：$\hat{R}_{ui}$——根据本章 RBPT 算法获得的预测评分；

μ——数据集的总体偏置；

bu——各个用户的偏置向量；

bi——各个项目的偏置向量。

对式(9-11)求偏导得到式(9-13)。

$$\begin{cases}
\dfrac{\partial L_T}{\partial U_i}=\displaystyle\sum_{j=1}^{M}I_{ij}^{R}g'(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V)[g(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]V_j+\beta\sum_{d\in \mathrm{Trust}(i)}T_{id}(U_i-U_d)+\beta\sum_{t\in \mathrm{Trusted}(i)}T_{ti}(U_t-U_i)+\lambda_U U_i \\
\dfrac{\partial L_T}{\partial V_j}=\displaystyle\sum_{i=1}^{N}I_{ij}^{R}g'(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V)[g(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]U_i+\lambda_V V_j \\
\dfrac{\partial L_T}{\partial bu(i)}=\displaystyle\sum_{i=1}^{N}I_{ij}^{R}g'(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V)[g(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]+\lambda_{bu}bu(i) \\
\dfrac{\partial L_T}{\partial bi(j)}=\displaystyle\sum_{j=1}^{M}I_{ij}^{R}g'(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V)[g(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]+\lambda_{bi}bi(j) \\
\dfrac{\partial L}{\partial \mu}=\displaystyle\sum_{j=1}^{M}I_{ij}^{R}g'(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V)[g(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)-R_{ij}]
\end{cases} \tag{9-13}$$

9.4.2 算法流程

算法基本流程如算法 9-1 所示。

算法 9-1：RBPT 算法

输入：用户—项目评分矩阵 R_{NM}，用户项目的潜在因子矩阵 U_{kN} 和 V_{kM}，可能的潜在因子维度 kVector，可能的最优信任权重 betaVector，算法最大迭代次数 maxEpoch，前后均方根误差(Root Mean Square Error，RMSE)阈值 Threshold；同时，为了减少模型复杂度设定随机梯度下降的学习速率，观察目标函数的变化情况来增大或者减小学习速率。

输出：用户潜在因子矩阵 U 和项目潜在因子矩阵 V，用户偏置 bu 和项目偏置 bi，最优潜在因子维度 k，最优信任权重。

步骤 1：使用正态分布初始化 U_{kN} = randn(k,N) * 0.01 和 V_{kM} = randn(k,M) * 0.01，其中乘以 0.01 是为了防止一开始 U、V 的内积太大，加快算法收敛速度；

步骤 2：针对实验所用数据集初始化 kVector=30：5：120(表示 k 从 30 遍历到 120，以 5 为间隔)，初始化 betaVector=[0.001 0.01 0.1 1 10 100](表示 beta 从 0.001 遍历到 100，每次扩大 10 倍)；

步骤 3：根据 R_{NM} 计算初始化得到数据集的总体偏置 μ；

步骤 4：初始化用户偏置向量 bu=randn(N,k) * 0.01 和项目偏置向量 bi=randn(M,k) * 0.01；

步骤 5：根据式(9-11)，结合 SGD 进行训练，参数的迭代过程如式(9-14)所示；

for 循环迭代直到最大迭代次 maxEpoch do

随机选择一条评分记录，按式(9-14)进行更新；

$$\begin{cases} U = U - \alpha \cdot \dfrac{\partial L}{\partial U} \\ V = V - \alpha \cdot \dfrac{\partial L}{\partial V} \\ bu = bu - \alpha \cdot \dfrac{\partial L}{\partial bu} \\ bi = bi - \alpha \cdot \dfrac{\partial L}{\partial bi} \\ \mu = \mu - \alpha \cdot \dfrac{\partial L}{\partial \mu} \end{cases} \tag{9-14}$$

真实评分数据索引 $R_L = R_{NM} > 0$；

计算均方根误差 RMSE=norm(($U^{T}V$-R_{NM}) * R_L,'Fro')/sqrt(sum(R_L))；

if 上一次和下一次迭代的 RMSE 大于 Threshold

continue；

else

算法结束；

end if

end for

步骤 6：根据每次选择的 k 和绘制 RMSE 随着迭代次数的变化曲线，选定 RMSE 最小时的 k 作为最优值

算法结束

9.4.3 算法复杂度分析

对式(9-11)进行计算的复杂度为 $O(\rho_R k + m u_T d)$，ρ_R 为 R 中已评分元素的个数，u_T 为一个用户平均信任的用户个数。梯度下降的复杂度为 $O(\rho_R k^2 + m(u_T + u'_T)d)$、$O(\rho_R k^2)O(\rho_R k^2)$、$O(\rho_R k^2)$ 和 $O(\rho_R k^2)O(\rho_R k^2)$，其中 $u'_T u'_T$ 表示一个用户平均被信任的用户个数。

由于用户在互联网中评分记录和信任记录服从幂律分布，长尾上的用户往往只有很少的用户信任数目，那么 $m u_T \ll \rho_R m u_T \ll \rho_R$；同理，$m u'_T \ll \rho_R m u'_T \ll \rho_R$，那么算法最终的复杂度是 $O(\rho_R k + 3\rho_R k^2)O(\rho_R k + 3\rho_R k^2)$，可以看出本章提出的 RBPT 算法与 ρ_R 线性相关，适用于大数据集。

9.5 实验分析

本节首先介绍实验中的数据来源和评价指标，然后介绍实验参数的确定，最后进行对比实验，并分析实验结果。为了不失一般性，随机选择 90%作为训练数据，然后预测余下的 10%的推荐精度，做 5 折交叉验证，取平均值。

本章的实验主要解决如下几个问题：

(1) 潜在因子的维度对推荐质量的影响；

(2) 用户项目偏置信息对推荐精度的影响；

(3) 潜在因子维度对推荐质量的影响；

(4) 社交网络信息对推荐结果的影响；

(5) 信任因子如何影响推荐质量；

(6) 社交网络正则化对推荐质量的影响；

(7) 和其他矩阵分解结合信任的推荐算法做比较。

不失一般性，为了降低模型复杂度，本章实验统一设置正则化因子 $\lambda_U = \lambda_V = \lambda_{bu} = \lambda_{bi} = 0.01$，设置 SGD 的学习速度 $\alpha = 0.03$。

9.5.1 实验数据集

为了充分地验证 RBPT 算法的性能，同时考虑到内存限制，本章选用两个数据挖掘领域公开可用的数据集——Ciao 和 FilmTrust。Ciao 数据集是由 JiliangTang 等人在 2012 年挖掘于网上社区，该数据集是“Ciao is a multi-million-strong online community”的递归缩写，评分范围是 1～5 的整数取值，评分间隔为 1 分。FilmTrust 数据集由 Golbeck 等在 2009 年挖掘于 FilmTrust 网站，评分为 1(极差)到 10(极好)，以 0.5 分为间隔。数据集是数据挖掘和社交网络推荐中广泛使用的数据集，包括用户项目评分信息和用户之间的信任关系。数据集的具体信息如表 9-1 所列。

表 9-1　实验所用数据集信息

数据集	用户数目	项目数目	评分记录数	用户评价电影数目		信任记录数	用户信任用户数目	
				最大值	最小值		最大值	最小值
Ciao	7375	106797	284086	1551	4	111781	804	0
FilmTrust	1642	2071	35497	244	1	1853	59	0

为什么选用这两个数据集？因为这两个数据集的评分矩阵稀疏度分别为0.036%和1.04%，相对于Movielens100k(943 1682 100000)的6.30%来说更加稀疏。

9.5.2　实验评价标准

为检验本章提出算法的推荐质量，实验采用均方根误差(Root Mean Square Error，RMSE)作为度量标准，如式(9-15)所示，这也是目前最常用的一种推荐质量度量方法，通过计算预测的用户评分与实际的用户评分之间的误差平方和的均方根来表示预测的准确性，RMSE值越小，推荐质量就越好。

$$\text{RMSE}=\sqrt{\frac{\sum_{u,i}(\hat{R}_{ui}-R_{ui})^2}{N}} \tag{9-15}$$

式中：$\hat{R}_{ui}$——预测的评分；

R_{ui}——测试集中的实际评分；

N——测试集包含的数据条数。

9.5.3　对比算法

为了验证本章提出算法的有效性，对偏置项和社交网络正则化进行验证，下面对本章涉及的对比实验进行说明。如式(9-16)所示是带偏置信息的概率矩阵分解，称之为BiasPMF算法，通过将BiasPMF算法与PMF算法进行对比来验证加入偏置项信息的有效性。

$$\begin{aligned}L(U,V,bu,bi)=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}\ [R_{ij}-g(\mu+bu(i)+bi(j)+U_i^{\mathrm{T}}V_j)]^2+\\&\frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|_{\mathrm{Fro}}^2+\frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|_{\mathrm{Fro}}^2+\frac{\lambda_{bu}}{2}\sum_{i=1}^{N}\|bu(i)\|_{\mathrm{Fro}}^2+\\&\frac{\lambda_{bi}}{2}\sum_{j=1}^{M}\|bi(i)\|_{\mathrm{Fro}}^2\end{aligned} \tag{9-16}$$

如式(9-17)所示是带有社交网络正则化但是无偏置项的目标函数，称为SocialReg算法。将该算法与PMF算法作对比，同时将BiasPMF与RBPT算法作对比来验证加入社交网络正则的作用。

$$L(R,T,U,V)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}^{R}(R_{ij}-g(U_i^{\mathrm{T}}V))^2+\frac{\beta}{2}\sum_{i=1}^{N}\sum_{d\in \mathrm{Trust}(i)}T_{id}\|U_i-U_d\|_F^2+\frac{\lambda_U}{2}\|U\|_F^2+\frac{\lambda_V}{2}\|V\|_F^2 \tag{9-17}$$

9.5.4 潜在因子维度的影响

如图 9-2 所示为不同数据集上 RBPT 算法随着潜在因子维度的变化情况，图 9-2(a)是 Caio 数据集中 RMSE 随潜在因子维度的变化情况，图 9-2(b)是 FilmTrust 数据集中 RMSE 随着潜在因子维度的变化情况。从图中可以看出，RBPT 算法关于不同的潜在因子维度其 RMSE 的变化比较大，说明在算法推荐过程中，数据集的维度选择对于推荐算法的推荐精度起着至关重要的作用；同时，从图中可以看出，RMSE 随分解维度的变化基本处于先下降后上升的趋势，从另一个侧面也反映了在解决实际问题过程中对推荐算法本身的要求也比较高。最终，选择 Ciao 数据集的潜在因子维度为 60，FilmTrust 数据集的潜在因子维度为 120。

另外，从图 9-2 可以看出，并不是维度越大推荐精度就越高，这是因为矩阵分解模型假定只有若干要素会影响到用户项目的潜在因子向量，如果潜在因子维度较小，对用户项目评分矩阵信息的挖掘较浅，精度提升缓慢；相反，如果潜在因子维度太大，则会引入噪声，从而降低推荐精度，这在文献[29]中进行了讨论，从侧面说明了参数选择的重要性。

9.5.5 偏置的影响

为了验证用户项目偏置信息对推荐质量的影响，绘制出如图 9-3 所示，PMF 算法和 BiasPMF 算法随着迭代次数的变化曲线，图 9-3(a)图对应的是 Caio 数据集，图 9-3(b)图对应的是 FilmTrust 数据集。

一般来说，RMSE 越小，算法的推荐效果越好。就图 9-3(a)和图 9-3(b)整体而言，随着迭代次数的增加，加入偏置项后的推荐精度明显得到提升，这恰好符合预期结果。具体来说，在 Ciao 数据集中，如图 9-3(a)所示，在迭代次数小于 15 次时，PMF 算法和 BiasPMF 算法随着迭代次数增加 RMSE 迅速下降，而且此时 PMF 算法对应的 RMSE 小于 BiasPMF 算法，表明此时采用 PMF 算法比采用 BiasPMF 能获得更好的推荐效果；当迭代次数增加到 20 次后，BiasPMF 算法的推荐效果开始优于 PMF 算法；随着迭代次数的进一步增加，BiasPMF 算法对应的 RMSE 一直处于下降状态，直至迭代增加到 50 次，这期间 PMF 算法所对应的 RMSE 基本不变。与图 9-3(a)表现不同的是，在 FilmTrust 数据集中，如图 9-3(b)所示，整体来说，BiasPMF 算法所对应的 RMSE 低于 PMF 算法，而且迭代曲线差别较大，说明 BiasPMF 算法相对于 PMF 算法在推荐精度上提升显著。具体而言，当迭代次数

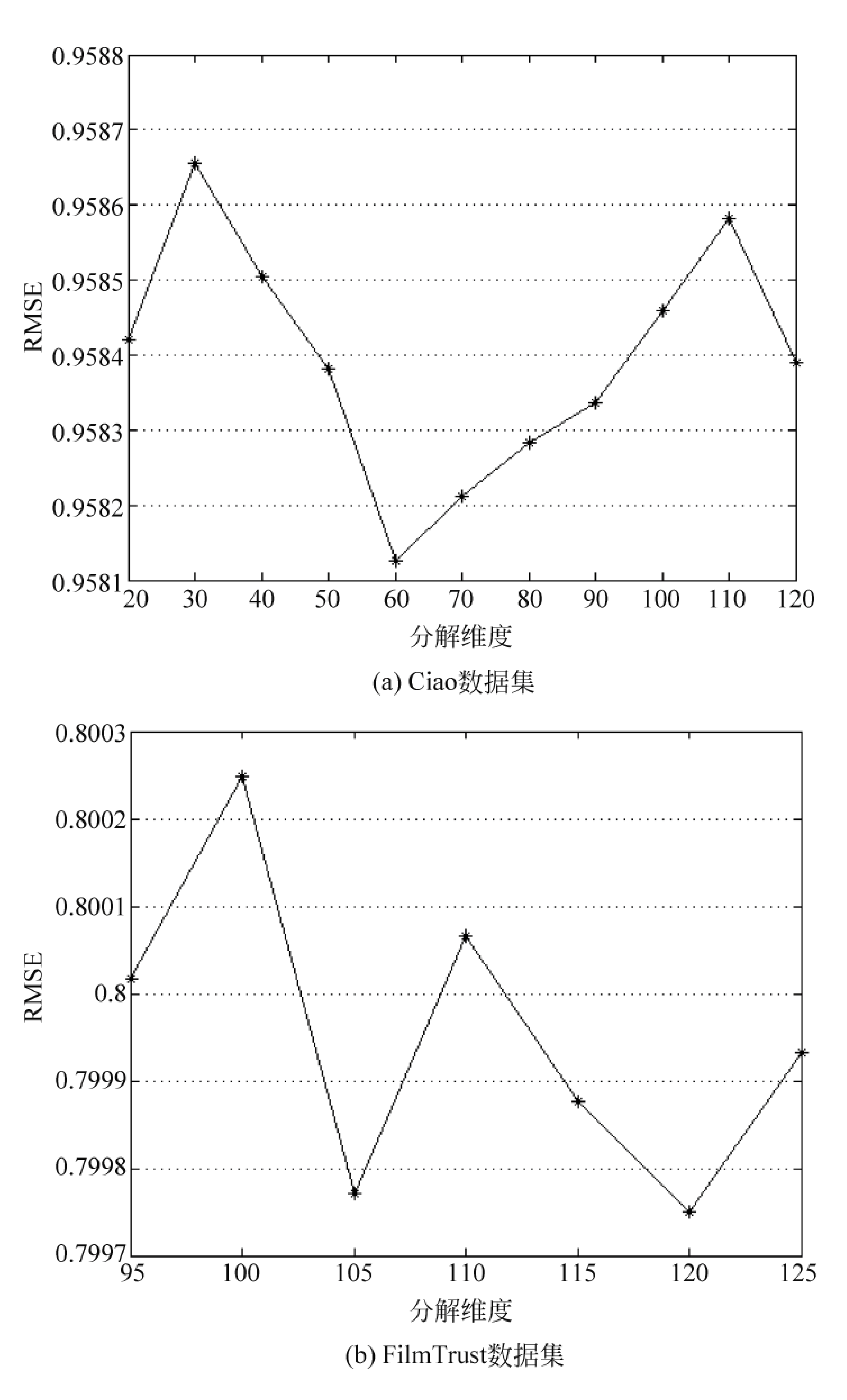
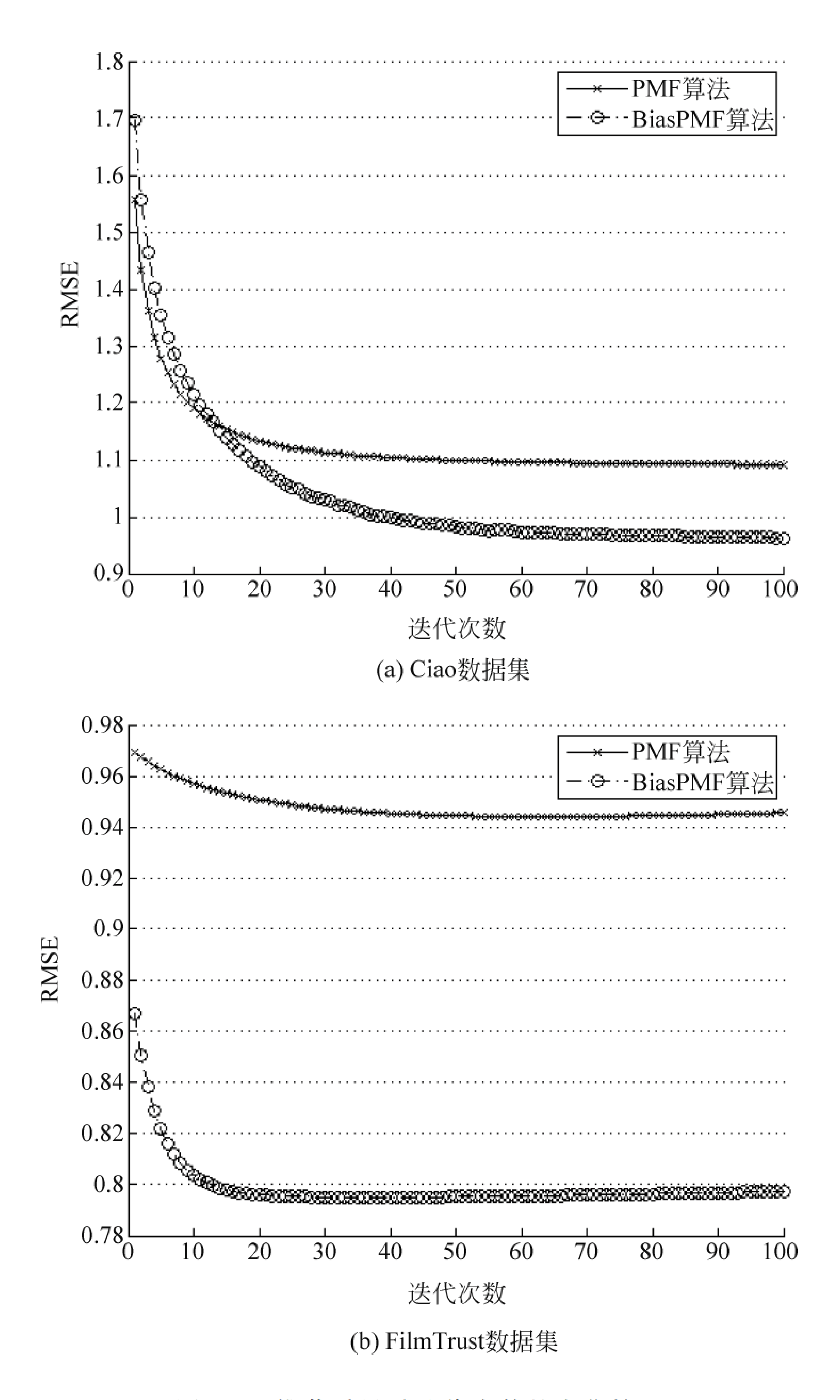


图 9-2　推荐质量随分解维度的变化情况

小于 20 次时，PMF 算法和 BiasPMF 算法所对应的 RMSE（Root Mean Square Error，均方根误差）一直处于下降状态，随着迭代次数的增加，RMSE 基本保持不变，只有轻微的过拟合现象，在 Ciao 数据集中并没有出现过拟合现象。

9.5.6　信任因子的影响

为了验证正则项对推荐质量的影响，进行如下实验。本实验包括两部分，第一部分是 PMF 算法与 SocialReg 算法的对比，第二部分是 BiasPMF 算法与 RBPT 算法的对比。



(a) Ciao数据集

(b) FilmTrust数据集

图 9-3 推荐质量随迭代次数的变化情况

1. PMF 与 SocialReg 算法对比

如图 9-4 所示是 PMF 算法和 SocialReg 算法在不同数据集上 RMSE 随着迭代次数的变化情况，图 9-4(a)是 Ciao 数据集上的实验，图 9-4(b)是 FilmTrust 数据集上的实验。从图 9-4 中可以看出，在 Ciao 数据集中，当迭代次数小于 20 次时，随着迭代次数的增加 PMF 算法和 SocialReg 算法所对应的 RMSE 均处于下降状态，而且 PMF 算法的推荐效果相对较好；当迭代次数在 20～80 次时，两种算法的推荐效果相差不大；随着迭代次数的进一步增加，SocialReg 算法的推荐效果开始

优于 PMF 算法。同样，在 FilmTrust 数据集中，随着迭代次数的增加 PMF 算法和 SocialReg 算法所对应的 RMSE 均处于下降态势，并且 SocialReg 算法的 RMSE 明显优于 PMF 算法。

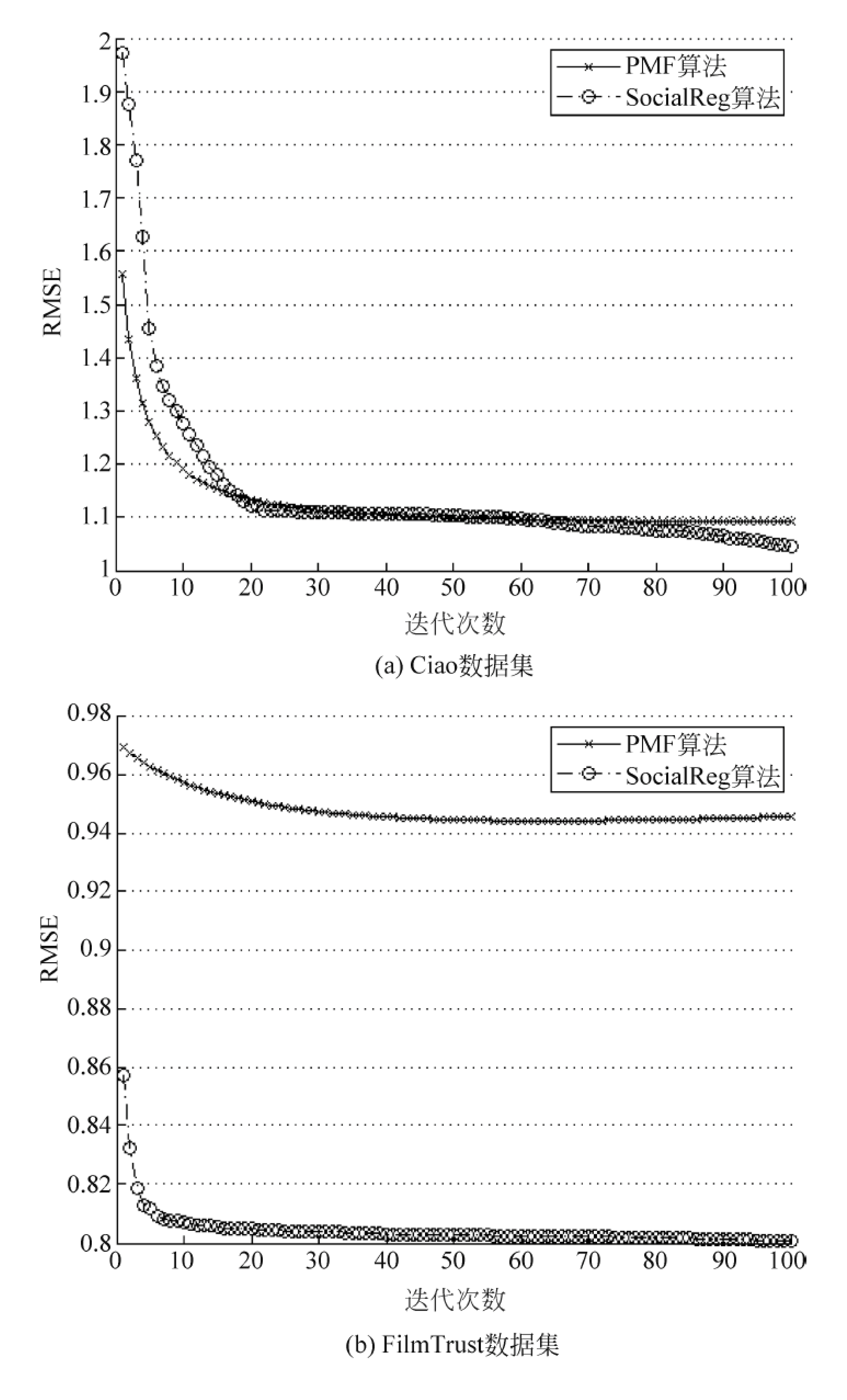


图 9-4　社交正则化对无偏置算法的影响

2. BiasPMF 算法与 RBPT 算法对比

如图 9-5 是 BiasPMF 算法和 RBPT 算法随迭代次数的变化情况。从图中可以看出，加入社交网络正则项的 RBPT 算法相对于未加入社交网络正则项的 BiasPMF 算法，不仅收敛速度快，而且在推荐精度上也有较大提升。

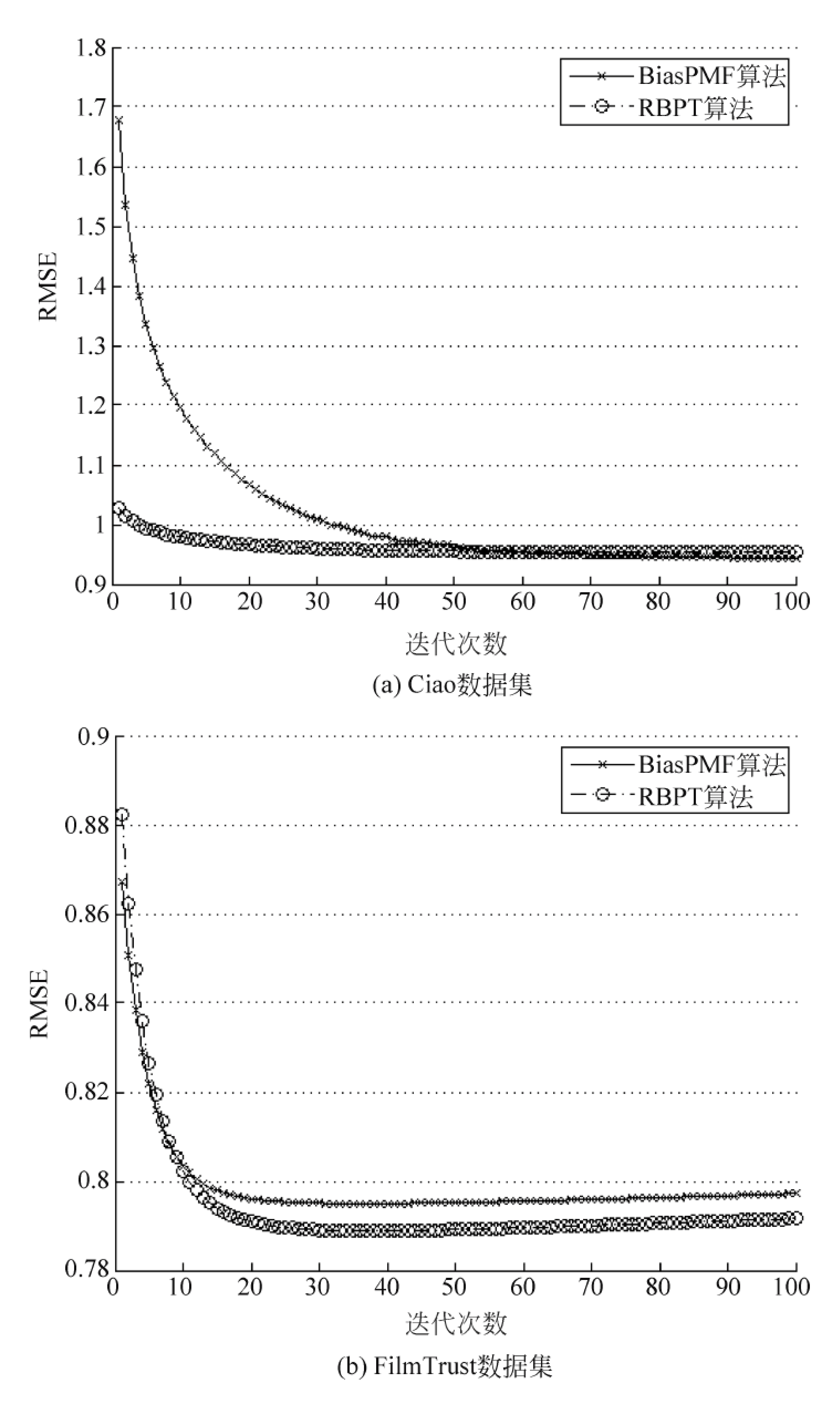


图 9-5 社交正则化对有偏置算法的影响

9.5.7 对比实验分析

如图 9-6 所示是不同算法的推荐质量随迭代次数的变化情况，包括 PMF 算法、BiasPMF 算法、SocialReg 算法和 RBPT 算法。对于不同的数据集，从图 9-6(a)和(b)可以看出，本章提出的 RBPT 算法均优于上述所提到的算法。具体来说，对于 Ciao 数据集，在 0～100 次的迭代过程中，RBPT 算法在第 10 次迭代时基本达到稳定状态，无论是迭代次数、还是 RMSE 的值都是明显优于其他三种算法。对于 FilmTrust 数据集，开始迭代时，所计算的 RMSE 值(0.88)处于比较高的状态(与

BiasPMF 算法、SocialReg 算法相比），但随着迭代次数的增加，其 RMSE 值几乎呈指数级的递减，说明由于数据集中用户之间存在信任关系，RBPT 算法起到了明显的优化作用，使 RMSE 值在迭代次数为 20 时基本达到稳定的最优值状态。对于 Ciao 数据集和 FilmTrust 数据集，详细的迭代次数和 RMSE 值之间的变化关系如表 9-2 所列，当达到稳定状态时，从表 9-2 可计算得到 RBPT 算法在 Ciao 数据集上相对于传统的 PMF 算法精度提升为 13.61%，在 FilmTrust 数据集上精度提升 15.52%，提升效果显著，说明本章提出的 RBPT 算法在 RMSE 方面具有一定的优越性，从而缓解了数据稀疏性带来的推荐精度不高的问题。

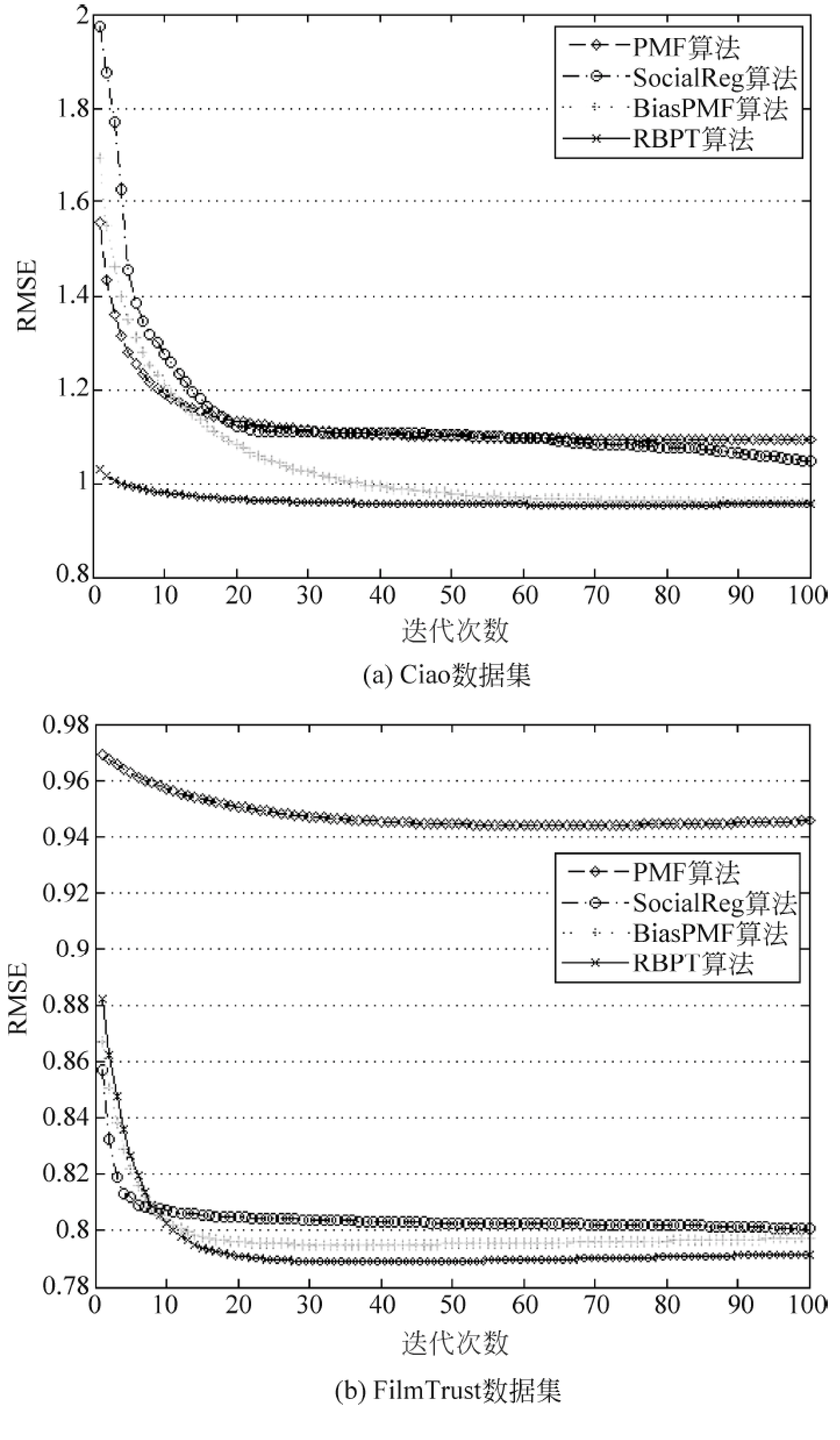


图 9-6　不同算法的推荐质量随迭代次数的变化情况

表 9-2 Ciao 和 FilmTrust 数据集中不同迭代次数对应的 RMSE 值

数据集	迭代次数	PMF 算法	SocialReg 算法	BiasPMF 算法	RBPT 算法
Ciao	1	1.5570	1.9735	1.6917	1.0284
	10	1.1902	1.2768	1.2097	0.9793
	20	1.1328	1.1211	1.0833	0.9662
	30	1.1131	1.1092	1.0247	0.9602
FilmTrust	1	0.9693	0.8572	0.8671	0.8821
	10	0.9570	0.8070	0.8035	0.8023
	20	0.9506	0.8047	0.7951	0.7909
	30	0.9472	0.8038	0.7950	0.7891

本章小结

本章提出一种基于社交网络正则化的改进概率矩阵分解算法，该模型基于现实世界中用户的消费行为不仅与自己的喜好有关，还与朋友的推荐有关，更加真实地反映了客观情况。实验结果表明本章提出的 RBPT 算法在两个数据集上的 RMSE 提高 13%～15%，同时算法的理论复杂性分析表明本章提出的算法随着数据规模线性变化，适用于大数据集。但是如何利用信任的传播与聚合来提升推荐精度将是接下来的需要深入研究的工作。

参考文献

[1] Ibrahim O. Collaborative filtering recommender systems[C]//International Conference on Intelligent Systems Design & Applications. IEEE, 2015: 438-443.

[2] Abdillah O, Adriani M. Mining User Interests through Internet Review Forum for Building Recommendation System [C]//Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on. IEEE, 2015: 564-569.

[3] Guan H, Li H, Xu C Z, et al. Semi-sparse algorithm based on multi-layer optimization for recommendation system.[J]. Journal of Supercomputing, 2012, 66(3): 148-155.

[4] Linden B G, Smith B, York J. Amazon. com Recommendations Iem-to-item collaborative filtering[J]. IEEE Internet Computing, 2015, 4(1): 76-80.

[5] Nikulin V, Huang T H, Ng S K, et al. A very fast algorithm for matrix factorization[J]. Statistics & Probability Letters, 2011, 81(7): 773-782.

[6] Andriy Mnih, Ruslan Salakhutdinov. Probabilistic Matrix Factorization[C]//Proc. of NIPS 2007, Vancouver, Canada, December, 2007, 1257-126.

[7] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 880-887.

[8] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008: 426-434.

[9] Rendle S. Factorization Machines with libFM[J]. ACM Transactions on Intelligent Systems & Technology, 2012, 3(3): 219-224.

[10] Guo G, Zhang J, Sun Z, et al. Librec: A Java library for recommender systems[C]// Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization, 2015: 955-963.

[11] Chin W S, Yuan B W, Yang M Y, et al. LIBMF: A Library for Parallel Matrix Factorization in Shared-memory Systems[J]. Technical Report, 2015: 32-37.

[12] Gueye M, Abdessalem T, Naacke H. A parameter-free algorithm for an optimized tag recommendation list size[J]. Recsys, 2014: 233-240.

[13] Niemann K, Wolpers M. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2013: 955-963.

[14] Yuan J, Li L. Recommendation based on trust diffusion model[J]. The Scientific World Journal, 2014, 2014(3): 159594-159594.

[15] Chen C, Zeng J, Zheng X, et al. Recommender system based on social trust relationships [C]//e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on. IEEE, 2013: 32-37.

[16] Moradi P, Ahmadian S, Akhlaghian F. An effective trust-based recommendation method using a novel graph clustering algorithm[J]. Physica A Statistical Mechanics & Its Applications, 2015, 436: 462-481.

[17] Zhou T, Shan H, Banerjee A, et al. Kernelized Probabilistic Matrix Factorization: Exploiting Graphs and Side Information[C]//SDM, 2012, 12: 403-414.

[18] Mehmet Gonen, Samuel Kaski. Kernelized Bayesian Matrix Factorization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(10): 2047-2060, 2014.

[19] Ma H, King I, Lyu M R. Learning to recommend with explicit and implicit social relations[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2 (3): 29.

[20] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009 (8): 30-37.

[21] Zulkefli N A M, Baharudin B. Travel recommendation system based on trust using hybrid neuro-fuzzy: a study of potential trust in blog and Facebook[J]. International Journal of Business Information Systems, 2015, 20(3): 289-309.

[22] Tang J, Gao H, Liu H, et al. eTrust: Understanding trust evolution in an online world [C]//Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2012: 253-261.

[23] Gemulla R, Nijkamp E, Haas P J, et al. Large-scale matrix factorization with distributed stochastic gradient descent[C]//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2011: 69-77.

[24] Hu X，Meng X，Wang L. Svd-based group recommendation approaches：an experimental study of moviepilot[C]//Proceedings of the 2nd challenge on context-aware movie recommendation. ACM，2011：23-28.

[25] Chin W S，Zhuang Y，Juan Y C，et al. A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems[J]. Acm Transactions on Intelligent Systems & Technology，2015，6(1)：1-24.

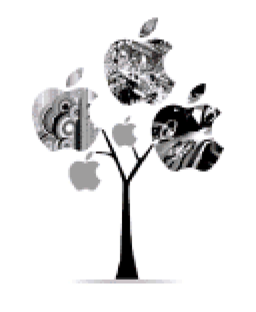
[26] Chin W S，Zhuang Y，Juan Y C，et al. A Learning-Rate Schedule for Stochastic Gradient Methods to Matrix Factorization[M]//Advances in Knowledge Discovery and Data Mining. Springer International Publishing，2015：442-455.

[27] Yang B，Lei Y，Liu D，et al. Social collaborative filtering by trust[C]//Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. AAAI Press，2013：2747-2753.

[28] Golbeck J. Generating Predictive Movie Recommendations from Trust in Social Networks [C]//. In：Proceedings of the 4th International Conference on Trust Management (iTrust'06)，Pisa，Italy. Springer-Verlag Berlin，Heidelberg，2006：93-104.

[29] Jamali M，Ester M. A matrix factorization technique with trust propagation for recommendation in social networks[C]//Proceedings of the fourth ACM Conference on Recommender Systems. ACM，2010：135-142.

第10章 带偏置的非负矩阵分解推荐算法



本章针对传统应用于推荐系统中的非负矩阵分解算法较少考虑独立于用户和项目之外的因素，提出一种结合用户及项目偏置的非负矩阵分解算法。为了避免随机初始化的用户—隐因子矩阵和项目—隐因子矩阵在更新过程中产生局部最优解，首先利用 SVD 技术初始化用户—隐因子矩阵和项目—隐因子矩阵。其次在分解过程中把用户、项目的偏置信息与传统非负矩阵分解算法相融合，明确偏置信息与预测数据之间的关系。最后通过实验表明，在不同的数据集上，该算法与传统矩阵分解算法相比在稀疏用户（评价项目比较少的用户）评分预测准确性上有显著提高。

10.1 引言

推荐系统的一个基本功能是利用用户—项目评分矩阵中已知评分预测未知评分，将预测评分高的项目推荐给用户。但是在研究过程中面临着诸多难题，其中最为典型的就是由数据的高维稀疏性带来的对稀疏用户（评价项目较少的用户）评分预测准确度不高的问题。

在现代化推荐系统中，用户、项目的数量通常都是呈指数级。随着项目数量的增加，用户评价的项目在总项目中所占比例越来越小，如果仍然采用适用于低维数据的相似度度量方法度量用户、项目之间的相似度，计算得到用户之间的相似区分度较低，随着数据量的增加，相似度的计算量将呈指数增长。针对稀疏矩阵中评分预测问题，近年来研究者们采用了各种各样的方式，Koren Y 提出把矩阵分解技术应用到推荐系统中，把用户—项目评分矩阵 $R_{m\times n}$ 分解为两个 k（$k\ll m,n$）维的用户—隐因子矩阵 $P_{m\times k}$ 和项目—隐因子矩阵 $q_{k\times n}$，利用分解后的隐因子矩阵预测评分，该算法与传统的协同过滤推荐算法相比在预测准确度上有比较大的提升，但是在分解后的矩阵中出现的负值与用户—隐因子、项目—隐因子原理不符。Lin C 提出利用每一个不为零的评分更新 $P_{m\times k}$、$q_{k\times n}$ 的 NMF（Non-negative Matrix Factorization）算法。NMF 算法在充分拟合已知评分的基础上对用户评分预测具有以下特点：

(1) 相似用户 m、t 分解后对应的行向量 $P_{m,\cdot}$ 与 $q_{t,\cdot}$ 相似度比较高；

(2) 相似用户 m、t 分解后对应的列向量 $q_{\cdot,m}$ 与 $q_{\cdot,t}$ 相似度比较高。

由以上分析可知，NMF 算法有助于提高相似用户的预测准确性，但在对稀疏用户评分预测方面表现一般。为了解决这一问题，本章在 NMF 算法的基础上，提出一种由 SVD 初始化，融合用户和项目偏执信息的非负矩阵分解算法，称为 RBNMF(Recommend with Bias Information NMF)算法，与传统矩阵分解算法相比 RBNMF 算法的稀疏用户预测准确有较大提高。

10.2 相关工作

10.2.1 矩阵分解

矩阵分解模型假设用户对项目的评分受到若干隐因子的影响，将用户和项目映射到一个共同的隐因子空间中，这种情况在实际工作中具体表现为：如果用户喜欢《C 语言程序设计》这本书，背后的原因可能是用户喜欢编程、用户喜欢谭浩强本人或用户对 C 语言比较感兴趣等，这背后的原因就称为隐因子。但是在矩阵分解过程中有些隐因子不能给出明确的解释，所以矩阵分解模型又称为隐语义模型。

矩阵算法是将用户—项目评分矩阵 $R_{m\times n}$ 分解成两个低维度的矩阵乘积，如式(10-1)所示。

$$R_{m\times n} \approx P_{m\times k} \times q_{k\times n} \tag{10-1}$$

式中：m——用户的个数；

n——项目的个数；

$k \ll \min(m,n)$——隐因子的个数。

假设评分误差服从高斯分布，采用梯度下降算法更新 $P_{m\times k}$、$q_{k\times n}$，为了防止过拟合现象的发生，可以在计算误差公式时加入正则项，如式(10-2)所示。

$$\min_{p^*,q^*} \sum_{(u,i)\in k} (R_{u,i} - P_{u,\cdot} \cdot q_{\cdot,i})^2 + \lambda(\| P_{u,\cdot} \|^2 + \| q_{\cdot,i} \|^2) \tag{10-2}$$

式中：K——已知评分；

$P_{u,\cdot}$——$R_{m\times n}$ 分解后 $P_{m\times k}$ 中的行向量；

$q_{\cdot,i}$——$R_{m\times n}$ 分解后 $q_{k\times n}$ 中的列向量；

λ——正则化因子。

基于矩阵分解的算法由于隐因子 $k \ll \min(m,n)$，实际计算的空间复杂度比较低；同时，该算法使用的是全局目标函数，所以预测准确度比较高。

10.2.2 奇异值矩阵

在推荐系统中，SVD 作为一种矩阵分解技术，通过分解产生低秩矩阵近似逼近原始矩阵。在一个给定的矩阵 A 中奇异值分解如式(10-3)所示。

$$A = U \times S \times V^T \tag{10-3}$$

式中：$U \in R_{m\times m}$，$V \in R_{n\times n}$，$S \in R_{m\times n}$；

矩阵 U 和矩阵 V——正交矩阵；

矩阵的列分别为 AA^{T}、$A^{\mathrm{T}}A$ 的特征向量；

矩阵 $S=\mathrm{diag}(\sigma_1,\sigma_2\cdots\sigma_r)$，$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r \geqslant 0$，其中 r 为矩阵 R 的秩；

σ_r——R 的奇异值。

SVD 算法可以分解为三个矩阵相乘对原始矩阵 A 进行最优逼近，通过将矩阵 S 保留前 K 个最大的奇异值进行化简得到更新后的对角矩阵，其中 $k \ll r$。然后通过删除 U 和 V 对应的列（U 和 V 中的列），化简后的矩阵 U 和 V 表示为 U_k 和 V_k。化简后 A' 可用式(10-4)表示。

$$A' = U_k \times S_k \times V_k^{\mathrm{T}} \tag{10-4}$$

10.2.3　Baseline 预测

测试算法性能好坏，往往需要建立一个对比基线，在对比的基础上观察后续试验效果的变化。观测到的评分数据有一些和用户无关的因素产生的效果，即一部分因素是和用户对物品的喜好无关而只取决于用户或物品本身的特性，例如，乐观的用户对于一些项目的评分普遍较高，而悲观的用户对项目的评分普遍较低，也就是说即使这两类用户对同一项目的评分相同，但是对物品的喜好程度确是不一样的。对于项目来说道理是一样的，受用户欢迎的项目评分普遍较高，不受用户欢迎的项目评分较低，加入偏执信息的评分预测公式如式(10-5)所示。

$$R^*(i,j) = \mu + bu(i) + bi(j) \tag{10-5}$$

式中：$R^*(i,j)$——用户 i 对项目 j 的预测评分；

μ——数据集的总体偏置信息；

$bu(i)$——用户 i 的偏置信息；

$bi(j)$——项目 j 的偏置信息。

假如项目 η 的总体偏置为 a，项目 η 的口碑普遍高于其他项目的值为 b，如果用户 U_1 是悲观严谨的，其 $bu(i)$ 值为 c，那么 U_1 对项目 η 的预测值为 $a+b-c$。

在计算偏置信息时，求解 $bu(i)$ 和 $bi(j)$ 的值如式(10-6)所示。

$$\begin{cases} bu(i) = \dfrac{\sum\limits_{u\in I}(R_{u,i} - \mu - b_i)}{\lambda_1 + |I|} \\ bi(j) = \dfrac{\sum\limits_{u\in Ui}(R_{u,i} - \mu)}{\lambda_2 + |U_i|} \end{cases} \tag{10-6}$$

式中：μ——所有已被评价项目的总体偏置；

u——用户；

i——项目；

I——用户 u 评价过的项目集合；

$|I|$——集合的个数；

U_i——评价过项目 i 的用户集合；

$|U_i|$——集合的个数；

参数 λ_1、λ_2 需要实验时确定。

10.2.4 NMF 算法

现代化推荐系统需要处理的数据量非常庞大，在现有矩阵分解的基础上 Lin C 提出了一种时间复杂度比较低的 NMF 算法，该算法利用每一个已知评分项更新分解后的用户—隐因子矩阵 $P_{m\times k}$ 和项目—隐因子矩阵 $q_{k\times n}$。在 Lin C 算法的基础上，为了防止分解后的矩阵出现过拟合，加入正则项的乘性迭代如式(10-7)所示。

$$\begin{cases} p_{u,k}=p_{u,k}\cdot\dfrac{\sum\limits_{i\in I_u}q_{k,i}\cdot r_{u,i}}{|I_u|\lambda_p p_{u,k}+\sum\limits_{i\in I_u}q_{k,i}\cdot r_{u,i}^*} \\ q_{k,i}=q_{k,i}\cdot\dfrac{\sum\limits_{u\in U_i}p_{u,k}\cdot r_{u,i}}{|U_i|\lambda_q q_{k,i}+\sum\limits_{i\in I_u}p_{u,k}\cdot r_{u,i}^*} \end{cases} \tag{10-7}$$

式中：I_u——评分不为零的项目集合；

U_i——评分不为零的用户集合；

$r_{u,i}$——用户 u 对项目 i 的实际评分；

$r_{u,i}^*$——预测的用户 u 对 i 的评分，其可以由初始化的用户—隐因子矩阵 $P_{m\times k}$ 和项目—隐因子矩阵 $q_{k\times n}$ 计算得到。

10.3 RBNMF 算法

10.3.1 理论分析

为了更好地利用 NMF 算法提升对稀疏用户评分的预测准确度，本章在误差函数中把用户、项目的偏置信息与用户—隐因子矩阵 $p_{m\times k}$ 和项目—隐因子矩阵 $q_{k\times n}$ 相结合，为了防止 $p_{m\times k}$、$q_{k\times n}$ 在训练开始阶段陷入局部最优解，利用 SVD 算法初始化 $p_{m\times k}$、$q_{k\times n}$。由上文可知融入了用户、项目偏置的损失函数可用式(10-8)所示。

$$\text{loss}=\frac{1}{2}\sum_{(u,i)\in R_k}((2r_{u,i}-\mu-bu-bi-p_{u,\cdot}\cdot q_{\cdot,i})^2+ \\ \frac{\lambda_p}{2}\cdot\|p_{u,\cdot i}\|^2+\frac{\lambda_q}{2}\cdot\|q_{u\cdot,i}\|^2) \tag{10-8}$$

式中：μ——项目的总体偏置；

bu——用户的偏置；

bi——项目 i 的偏置；

$p_{m\times k}$、$q_{k\times n}$——分解后的与用户和项目相关的矩阵；

λ_p，λ_q——正则化因子；

r_k——不为零评分项的集合。

由式(10-8)可以得 $p_{m\times k}$、$q_{k\times n}$的更新方程，如式(10-9)所示。

$$\begin{cases} p_{u,k} = p_{u,k} - \eta_{u,k} \cdot \dfrac{\partial \text{loss}}{\partial p_{u,k}} \\ q_{k,i} = q_{k,i} - \eta_{k,i} \cdot \dfrac{\partial \text{loss}}{\partial q_{k,i}} \end{cases} \tag{10-9}$$

式中：$\eta_{u,k}$——用户的每次更新步长；

$\eta_{k,i}$——项目的每次更新步长。

为了使用使分解后的矩阵符合隐因子原理，本章采用乘性迭代保证每次迭代的值都为正。

对式(10-9)进行如下改进：

(1) 把式(10-9)简化为式(10-10)；

(2) 为了迭代过程中出现乘性迭代，对于参数选择的主要目的是消掉相加的第一项，经过化简得到 $\eta_{u,k}$，$\eta_{k,}$，如式(10-11)所示。

$$\begin{cases} p_{u,k} = p_{u,k} + \eta_{u,k} \cdot \sum\limits_{u\in Ui} q_{k,i} \cdot 2R_{u,i} - \\ \qquad \eta_{u,k} \cdot \sum\limits_{u\in Ui} q_{k,i}(p_{u,\cdot} \cdot q_{\cdot,i} + \mu + bu + bi) + \lambda_p p_{u,k} \\ q_{k,i} = q_{k,i} + \eta_{k,i} \cdot \sum\limits_{i\in I_u} p_{u,k} \cdot 2R_{u,i} - \\ \qquad \eta_{u,k} \cdot \sum\limits_{i\in I_u} p_{u,k}(p_{u,\cdot i} \cdot q_{u\cdot,i} + \mu + bu + bi) + \lambda_q q_{k,i} \end{cases} \tag{10-10}$$

$$\begin{cases} \eta_{u,k} = \dfrac{p_{u,k}}{\sum\limits_{u\in Ui} q_{k,i}(p_{u,i} \cdot q_{u,i} + \mu + bu + bi) + \lambda_p p_{u,k}} \\ \eta_{k,i} = \dfrac{q_{k,i}}{\sum\limits_{i\in I_u} p_{u,k}(p_{u,i} \cdot q_{u,i} + \mu + bu + bi) + \lambda_q q_{k,i}} \end{cases} \tag{10-11}$$

式中：U_i——评价过项目的用户集合；

I_u——被用户评价过的项目集合。

将式(10-11)代入式(10-10)得到 $p_{u,k}$，$q_{k,i}$的乘性迭代公式，如式(10-12)、式(10-13)所示。

$$p_{u,k}=p_{u,k}\frac{\sum\limits_{u\in Ui}q_{k,i}\cdot 2R_{u,i}}{\sum\limits_{u\in Ui}q_{k,i}(p_{u,\cdot i}\cdot q_{u\cdot,i}+\mu+bu+bi)+\lambda_p p_{u,k}} \tag{10-12}$$

$$q_{k,i}=q_{k,i}\frac{\sum\limits_{i\in I_u}p_{u,k}\cdot 2R_{u,i}}{\sum\limits_{i\in I_u}p_{u,k}(p_{u,i}\cdot q_{u,i}+\mu+bu+bi)+\lambda_q q_{k,i}} \tag{10-13}$$

评分形成由四部分构成，即数据集总体偏置、用户偏置、项目偏置和预测值，如式(10-14)所示。

$$P_{\text{score}}=\frac{1}{2}(p_{u,i}\cdot q_{u,i}+\mu+bu+bi) \tag{10-14}$$

10.3.2 RBNMF 算法流程

为了防止 $p_{m\times k}$、$q_{k\times n}$在训练开始阶段陷入局部最优解，在产生初始化项目隐因子矩阵时采用 SVD 算法对实验数据集进行初步训练，得到左奇异矩阵 U、右奇异矩阵 V 及奇异矩阵 S，接着选择 S 中前 k 个的奇异值得到 $\sqrt{S(k)}$，那么初始前因子矩阵为 $P=U*\sqrt{S(k)}$，$Q=\sqrt{S(k)}*V^{\mathrm{T}}$。

算法 10-1：RBNMF 算法

输入：用户项目评分矩阵 $R_{m\times n}$，初始用户隐因子矩阵 $p_{m\times k}$，初始项目隐因子矩阵 $q_{k\times n}$，训练次数 Round=1000 次，初始化用户偏置向量 bu_m 和项目偏置向量 bi_n，初始正则化因子 λ_p，λ_q。初始化中间计算过渡矩阵 UP，UD，IP，ID。

输出：用户隐因子矩阵 $p_{m\times k}$，项目隐因子矩阵 $q_{k\times n}$，μ，bu_m，bi_n。

步骤 1：根据 SVD 算法训练得到 $U_{m\times k}$、$V_{k\times n}$ 和最优因子维度 K，然后利用 $P=U*\sqrt{S(k)}$，$Q=\sqrt{S(k)}*V^{\mathrm{T}}$ 得到初始化用户隐因子矩阵 $P_{m\times k}$ 及项目隐因子矩阵 $Q_{k\times n}$。

步骤 2：利用初始化的 $P_{m\times k}$、$Q_{k\times n}$ 计算 μ，由此利用式(10-6)计算出 bu_m，bi_n。

步骤 3：利用每一个不为零的评分项训练出中间矩阵值。

```
for (R_{i,j}≠0)  do
    将过渡矩阵 UP，UD，IP，ID 初始化为 0 矩阵。
    for (f=1; f<k; f++)
        UP_{m,f} = UP_{m,f} + Σ_{u∈Ui} q_{k,i} · 2R_{u,i};
        UD_{m,f} = UD_{m,f} + Σ_{u∈Ui} q_{k,i}(p_{u,i} · q_{u,i}+μ+bu+bi)+λ_p p_{u,k};
        IP_{f,n} = IP_{f,n} + Σ_{i∈I_u} p_{u,k} · 2R_{u,i};
        ID_{f,n} = ID_{f,n} + Σ_{i∈I_u} p_{u,k}(p_{u,i} · q_{u,i}+μ+bu+bi)+λ_q q_{k,i};
    end
end
for(每一个用户)
    for(f=1; f< k; f++)
```

续表

```
        P_{m,f} = P_{m,f} · UP_{m,f} / UD_{m,f};
        end
    end
    for(每一个项目)
        for(f=1; f<k; f++)
            Q_{f,n} = Q_{f,n}/IP_{f,n} / ID_{f,n};
        end
    end
```

步骤 4：利用测试集测试和一次训练生成的 $p_{m\times k}$、$q_{k\times n}$、bu_m、bi_n 以及式(10-16)计算 RMSE 值。

```
  if 前一次得到的 RMSE 值大于这次的值，
        进入下一次训练；
    else
        记录当前参数值；
    end
```

步骤 4：返回最优参数值及 $p_{m\times k}$、$q_{k\times n}$、μ、bu_m、bi_n。

算法结束

10.4　实验分析

本节首先介绍实验数据集的来源，然后介绍实验参数的确定，最后进行对比试验并分析实验结果。

本章进行的实验主要解决以下问题：

(1) 偏置参数的确定；

(2) 潜在因子维度对推荐质量的影响；

(3) 本章算法与 NMF 与 RBNMF 推荐算法对稀疏用户评分预测比较。

10.4.1　数据集

本章实验分别在 Epinion、Ciao、MovieLens 三个数据集进行，这三个数据集都包含了用户对项目的评分且分值分为 1～5 的离散值，数据集简介如表 10-1 所示。

表 10-1　数据集简介

数据集	用户数目	项目数目	评分数	稀疏度
MovieLens	6040	3706	1000209	4.47%
Ciao	7375	99746	278483	0.0379%
Epinion	40163	139738	664824	0.0118%

10.4.2 评价标准

为了检验本章提出算法的推荐质量，本实验采用均方根误差(Root Mean Square Error，RMSE)作为度量标准，通过计算预测值与实际值之间的 RMSE 来表示预测的准确性，RMSE 值越小，推荐质量就越好，如式(10-15)所示。

$$\text{RMSE} = \sqrt{\frac{\sum_{i}^{n}(x_i - x_o)^2}{n}} \tag{10-15}$$

式中：x_i——代表预测值；

x_o——与预测值对应的真实值。

10.4.3 实验结果及分析

首先将数据集的 90%作为训练集，其余的 10%作为测试集。选择 BaseLine 预测的目的在于该算法的训练时间短，预测精度高，可以通过实验训练得到最优参数。

如图 10-1 所示是 MovieLens 1M 数据集 Baseline 预测，经实验发现当 $\lambda_1=3$，$\lambda_2=6$，RMSE 达到最优，最小值为 0.961。

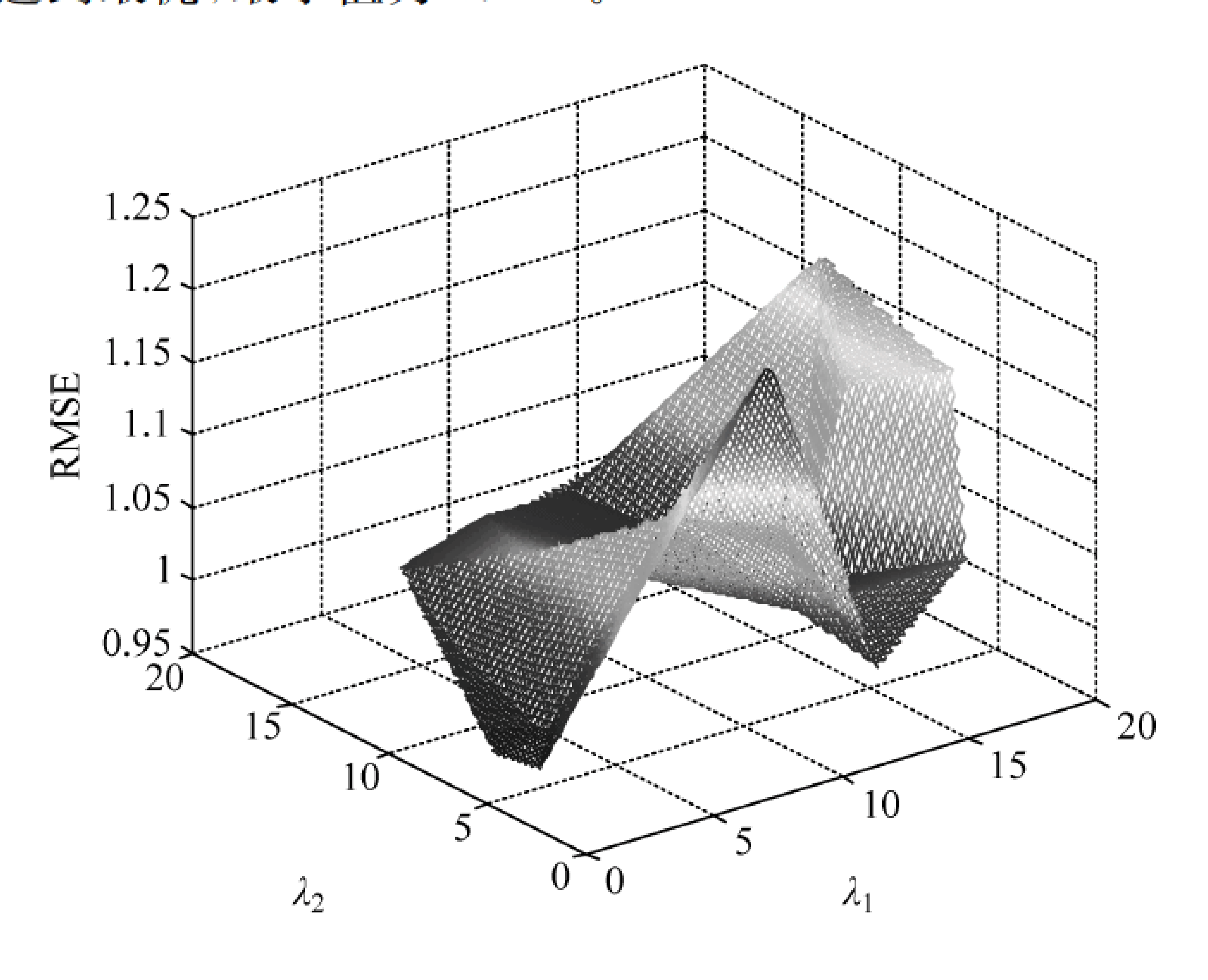


图 10-1 MovieLens 1M 数据集 Baseline 预测

如图 10-2 所示是 Ciao 数据集上所做的 Baseline 预测，经实验发现当 $\lambda_1=58$，$\lambda_2=41$，RMSE 达到最优，最小值为 0.976。

如图 10-3 所示是 Epinions 数据集上的 Baseline 预测，经实验发现当 $\lambda_1=55$，$\lambda_2=45$，RMSE 达到最优，最小值为 0.998。

由以上实验确定了参数值之后，在 MovieLens 1M 数据集上通过多次实验，参数 $\lambda_1=3$，$\lambda_2=6$，$\lambda_p=1.35$，$\lambda_q=0.96$ 时 RMSE 表现最优，此时对比 NMF 算法与本章提出的 RBNMF 算法在不同的维度上 RMSE 值，如图 10-4 所示为 MovieLens

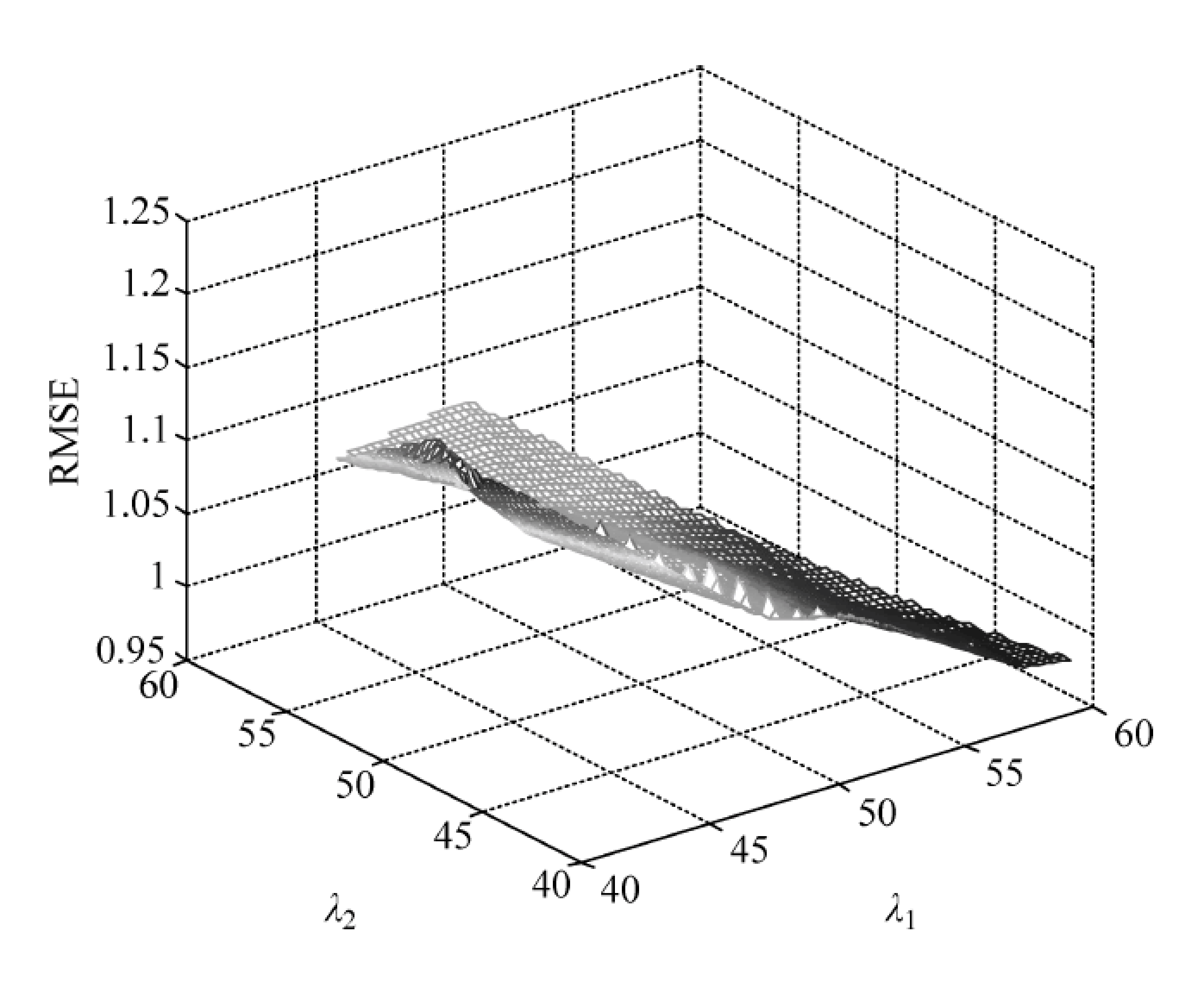


图 10-2　Ciao 数据集 Baseline 预测

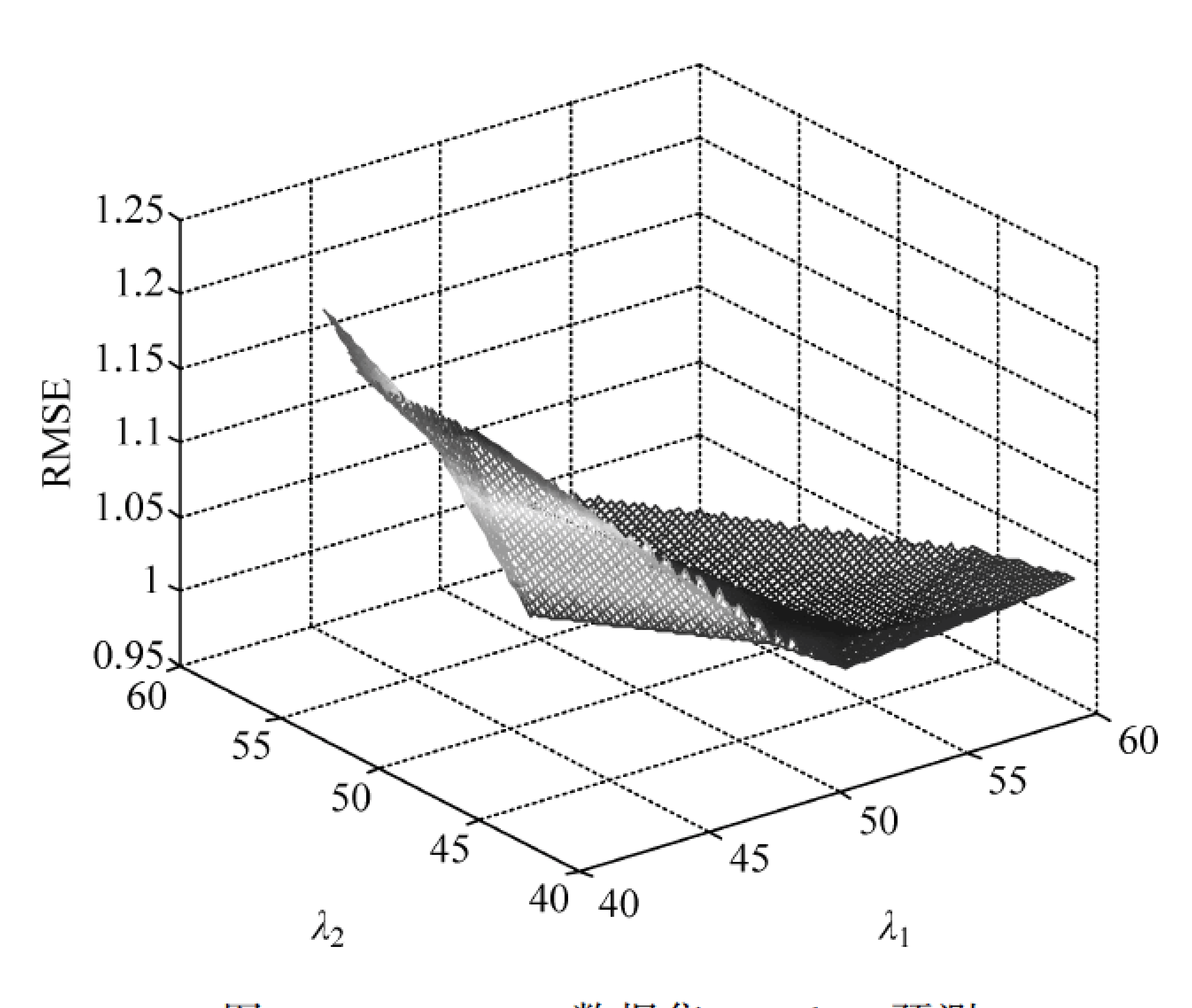


图 10-3　Epinion 数据集 Baseline 预测

1M 数据集分解不同维度 RMSE 对比。在 Ciao 数据集上通过多次实验，参数 $\lambda_1=58$，$\lambda 2=41$，$\lambda_p=1.62$，$\lambda_q=1.06$ 时 RMSE 表现最优，此时对比上 NMF 算法与本章提出的 RBNMF 算法在不同的维度上 RMSE 值，如图 10-5 所示为 Ciao 数据集分解不同维度 RMSE 对比。在 Epinion 数据集上通过多次实验，参数 $\lambda_1=55$，$\lambda_2=45$，$\lambda_p=1.73$，$\lambda_q=0.786$ 时 RMSE 表现最优，此时对比 NMF 算法与本章提出的 RBNMF 算法在不同维度上的 RMSE 值，如图 10-6 所示为 Epinion 数据集分解不同维度 RMSE 对比。可以明显看出，NMF 算法的 RMSE 值随着分解维度的增加，变化幅度与本章提出的算法相比变化范围较大。这说明了本章提出的 RBNMF 算法在训练过程中对分解维度的要求弱于 NMF 算法，也恰恰证明了本章算法通过添加用户、项目偏置信息，有力地改善了对于一些难以通过实验确定最佳

分解维度的用户评分预测。当分解维度到达一定值时在不同数据集上两种算法的最优 RMSE 基本相等，如果分解维度继续增加，NMF 算法及 RBNMF 算法都会显示出预测精确度下降的趋势，但是本章提出的算法改变幅度较小，原因是一旦隐因子数量超过一定范围，相对多增加的隐因子会在一定程度上削弱最佳隐因子（也即 RMSE 对应最低的维度数）对预测值的影响，实际表现就是预测精度的降低。

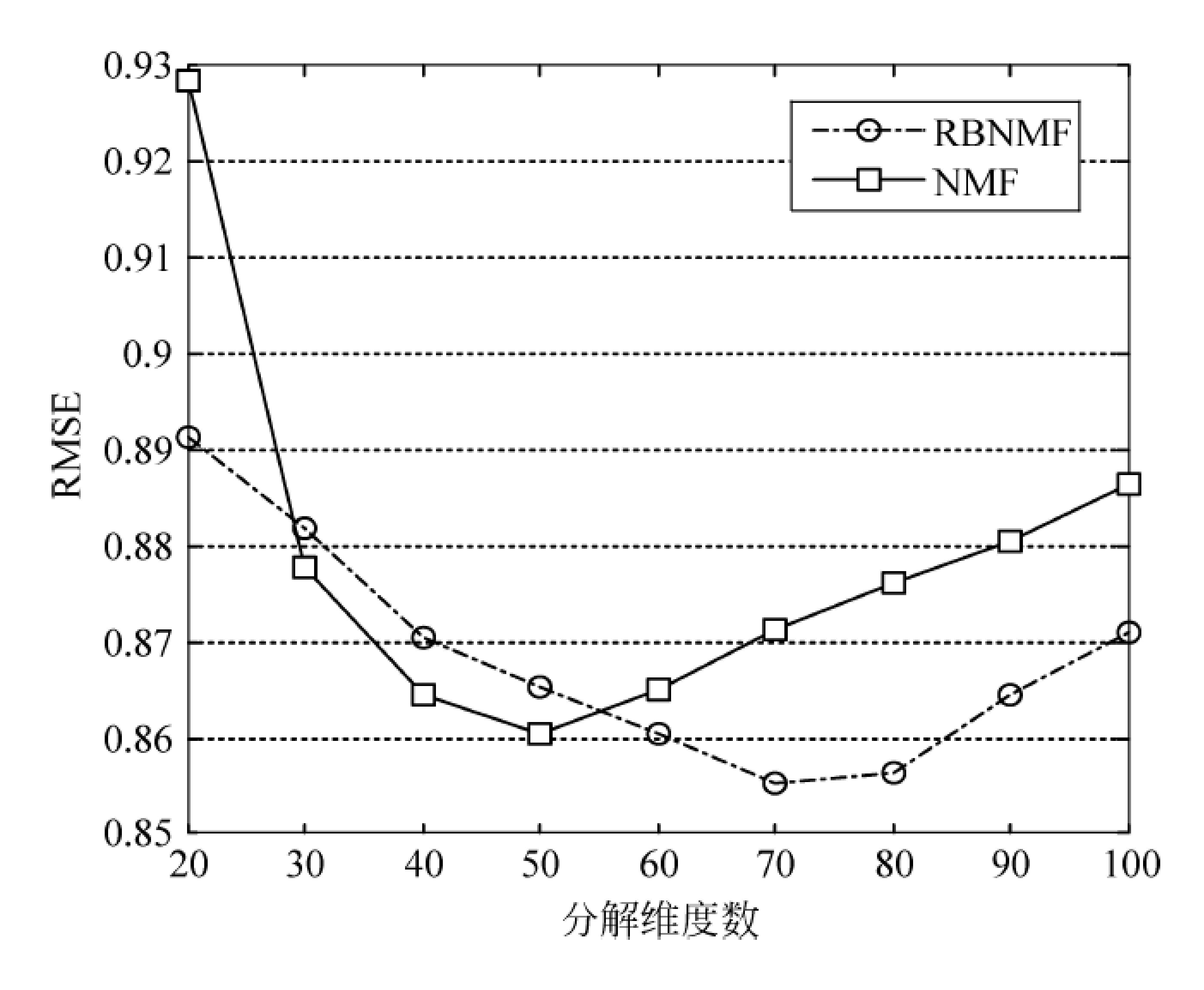


图 10-4　MovieLens 1M 数据集分解不同维度 RMSE 对比

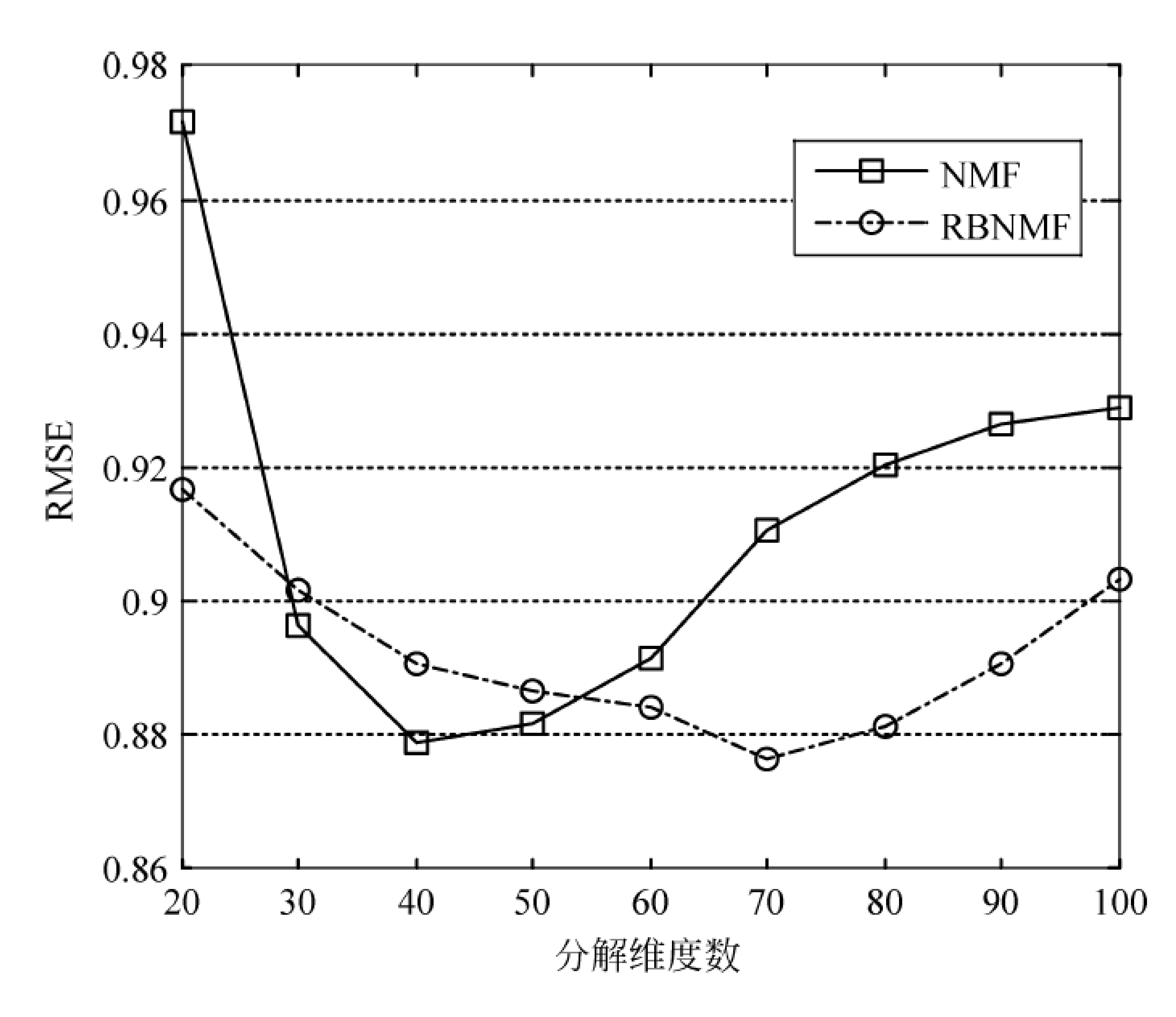


图 10-5　Ciao 数据集分解不同维度 RMSE 对比

对于大型数据集，稀疏用户的数量比较大。图 10-7 列出了 MovieLens 1M 数据集、Ciao 数据集、Epinion 数据集的用户评价项数量分布，可以看出稀疏用户（四类中的第一类）占全部用户的比例较大。

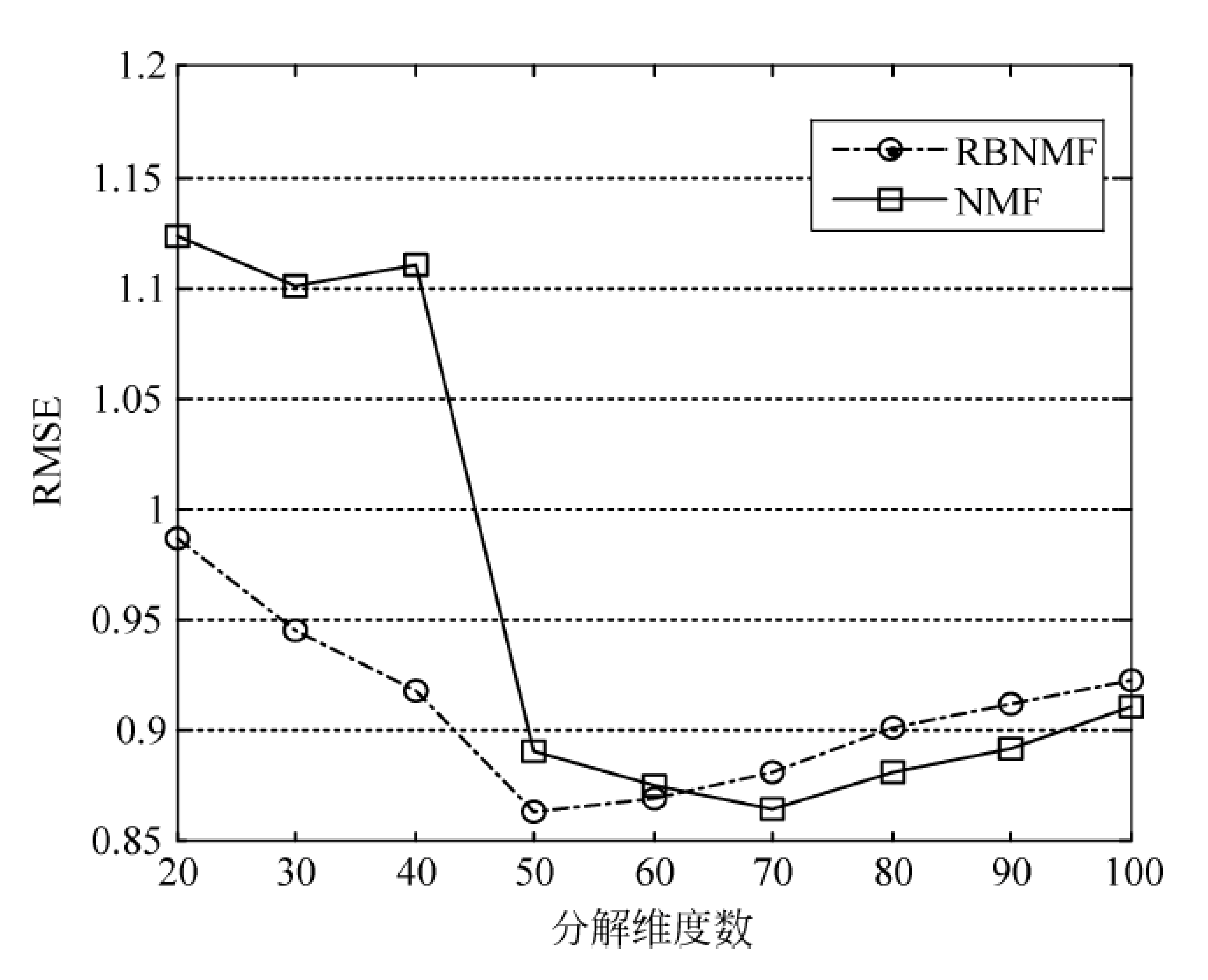


图 10-6　Epinion 数据集分解不同维度 RMSE 对比

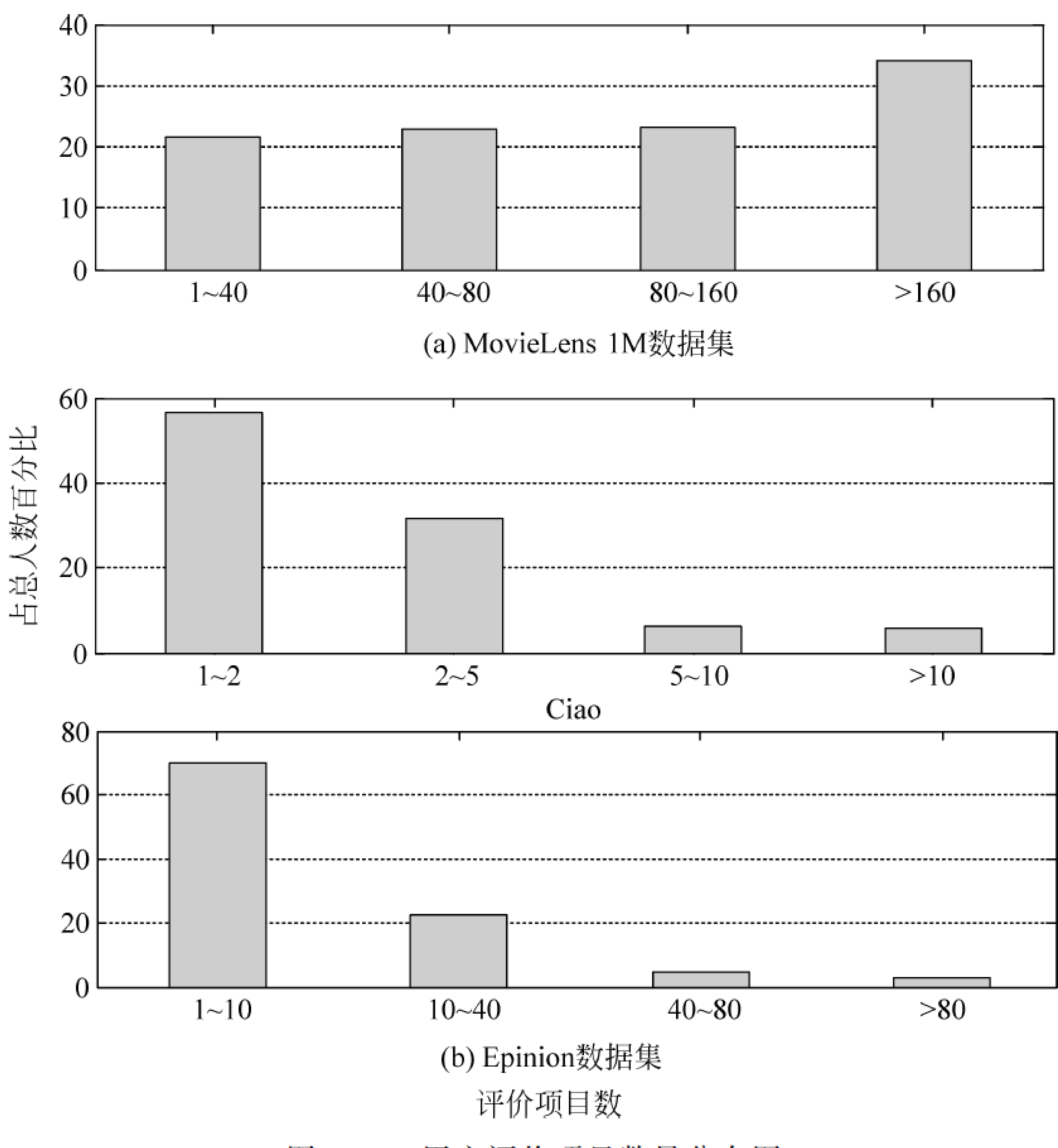


图 10-7　用户评价项目数量分布图

图 10-8 选择三种数据集的稀疏用户（图 10-7 中每种数据集的第一类用户）作为测试集，从上述实验中选取三种不同数据集达到最优的 λ_1、λ_2、λp、λ_q。可以看

出，在实验开始阶段，三种数据集在分解维度数相同时，本章提出的 RBNMF 算法的预测准确度优于 NMF 算法。针对稀疏性较大的 Ciao 和 Epinion 数据集的预测准确性，分解维度数为 20 时，相比于稀疏性较低的 MovieLens 数据集展现出比较大的优势，更加明确地说明了用户—项目偏置对稀疏用户的评分预测准确性有较大提升。随着分解维度的增加三种算法的预测准确性有明显提高，原因是隐因子模型随着挖掘的潜在因素的增加预测值趋于更加合理化。随着分解维度的进一步增加，相比各个数据集中全部用户，稀疏用户的预测准确度下降并不明显，原因是稀疏用户评价项目较少，约束隐因子条件较少，不能确定最优分解维度数（如 MovieLens 数据集中最优维度数可以为 50～100 的任何一个）。

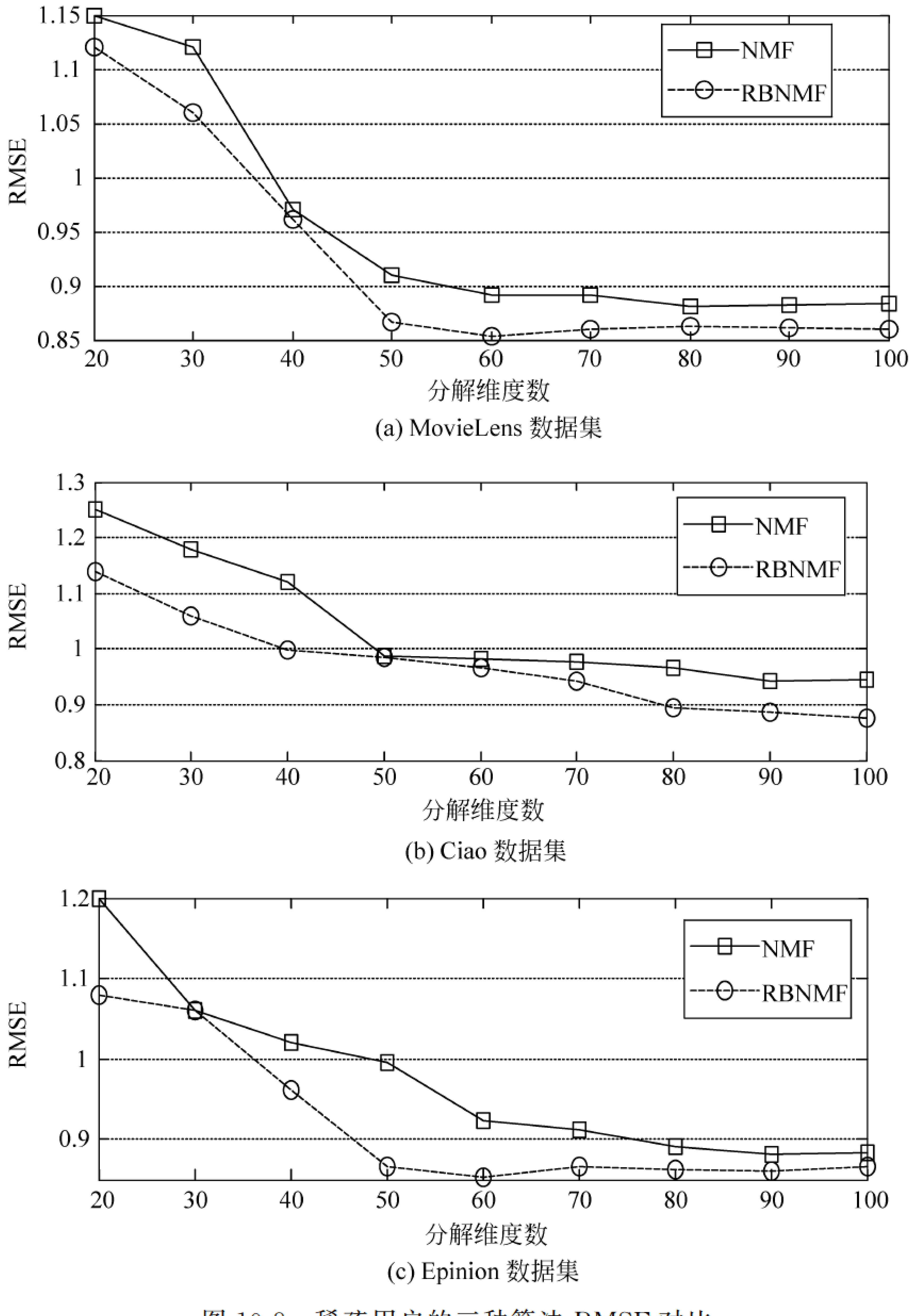


图 10-8 稀疏用户的三种算法 RMSE 对比

本章小结

本章研究了非负矩阵分解算法在推荐系统中的应用，RBNMF 算法在 NMF 的基础上加入了独立于观测的评分数据和用户无关的偏置信息。实验表明，对比传统的矩阵分解算法，本章提出的算法针对稀疏用户评分预测精确性有较大的提高。但是，在算法的改进过程中充分利用已知评分的机制还不够完善，进一步研究的重点应该是不同用户之间信任的建立、传播，把信任机制引入到矩阵分解算法中。

参考文献

[1] Herlocker J L, Konstan J A, Riedl J. Explaining collaborative filtering recommendations [A]. P Hinds. ACM Conference on Computer Supported Cooperative Work. ACM, 2001: 5-53.

[2] Deshpande M, Karypis G. Item-based top-*N* recommendation algorithms[J]. Acm Transactions on Information Systems, 2004, 22(1): 143-177.

[3] Cheung K W, Tian L F. Learning User Similarity and Rating Style for Collaborative Recommendation[J]. Information Retrieval Journal, 2004, 7(3): 395-410.

[4] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems[J]. Computer, 2009, 42(8): 30-37.

[5] Hoyer P O. Non-negative Matrix Factorization with Sparseness Constraints[J]. Journal of Machine Learning Research, 2004, 5(1): 1457-1469.

[6] Lin C. Projected Gradient Methods for Nonnegative Matrix Factorization[J]. Neural Computation, 2007, 19(10): 2756-79.

[7] Aharon M, Elad M, Bruckstein A. SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation[J]. IEEE Transactions on Signal Processing, 2006, 54(11): 4311-4322.

[8] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.

[9] Miller F P, Vandome A F, Mcbrewster J. Gradient Descent[J]. Alphascript Publishing, 2010, 20(4): 235-242.

[10] Osher S, Yin W, Goldfarb D, et al. An Iterative Regularization Method for Total Variation-Based Image Restoration[J]. Siam Journal on Multiscale Modeling & Simulation, 2005, 4(2): 460--489.

[11] Dan K. A singularly valuable decomposition: The SVD of a matrix[J]. College Mathematics Journal, 1996, 27(1): 2--23.

[12] Marinho L B, Hotho A, Jäschke R, et al. Baseline Techniques[M]. US: Springer US, 2012.

[13] Tang Z, Zhang X, Zhang S. Robust Perceptual Image Hashing Based on Ring Partition and NMF[J]. IEEE Transactions on Knowledge & Data Engineering,

2014，26(3)：711-724.

[14] Nguyen G T，Ahn H. A Combining Method of Content-based Information into Matrix Factorization Recommendation System[J]. 2016，53：204-218.

[15] Pirasteh P，Hwang D，Jung J J. Exploiting matrix factorization to asymmetric user Similarities in recommendation systems[J]. Knowledge-Based Systems，2015，83(1)：51-57.

[16] Gomez-Uribe C A，Hunt N. The Netflix Recommender System：Algorithms，Business Value，and Innovation[J]. Acm Transactions on Management Information Systems，2016，6(4)：13.

[17] Rampure V，Tiwari A. A Rough Set Based Feature Selection on KDD CUP 99 Data Set[J]. International Journal of Database Theory & Application，2015，8.

[18] 赵恒. 基于 LBS 的本地美食推荐系统的研究与实现[D]. 成都：电子科技大学，2015.

[19] Matuszyk P，Vinagre J，Spiliopoulou M，et al. Forgetting methods for incremental matrix factorization in recommender systems[C]//ACM Symposium on Applied Computing. ACM，2015：947-953.

[20] Zhao C，Peng Q，Zhang Z. A Matrix Factorization Algorithm with Hybrid Implicit and Explicit Attributes for Recommender Systems[J]. Journal of Xian Jiaotong University，2016.

[21] 陈克寒，韩盼盼，吴健. 基于用户聚类的异构社交网络推荐算法[J]. 计算机学报，2013，36(2)：349-359.

[22] 金淳，张一平. 基于 Agent 的顾客行为及个性化推荐仿真模型[J]. 系统工程理论与实践，2013，33(2)：463-472.

[23] Lin H，Yang X，Wang W，et al. A Performance Weighted Collaborative Filtering algorithm for personalized radiology education[J]. Journal of Biomedical Informatics，2014，51：107.

[24] 乌达巴拉，汪增福. 基于半监督的短语情感倾向性分析方法[J]. 模式识别与人工智能，2016，29(4)：289-297.

[25] 张锋，常会友. 使用 BP 神经网络缓解协同过滤推荐算法的稀疏性问题[J]. 计算机研究与发展，2015，43(4)：667-672.

[26] Boutet A，Frey D，Guerraoui R，et al. Privacy-preserving distributed collaborative filtering[J]. Computing，2016，98(8)：827-846.

[27] Liu J，Tang M，Zheng Z，et al. Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation[J]. IEEE Transactions on Services Computing，2016：1-1.

[28] Nasi R，Taber A，Vliet N V. Empty Forests，Empty Stomachs? Bushmeat and Livelihoods in the Congo and Amazon Basins[J]. International Forestry Review，2016，13(3)：14.

[29] Sampoornam M. Collaborative Based Filtering Approach for Web Service Recommendations using GEOLocations[J]. 2015，3(3)：1045-1047.

[30] Jr E C T，Ferrucci P，Duffy M. Facebook use，envy，and depression among college students：Is facebooking depressing? [J]. Computers in Human Behavior，2015，43

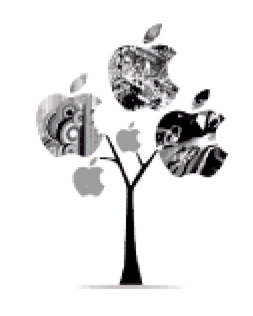
(43)：139-146.

[31] Kingsbury B E D，Sainath T N，Sindhwani V. Low-rank matrix factorization for deep belief network training with high-dimensional output targets[J]. 2016：6655-6659.

[32] 涂丹丹，舒承椿，余海燕. 基于联合概率矩阵分解的上下文广告推荐算法[J]. 软件学报，2013(3)：454-464.

[33] 刁海伦. 基于社交网络的个性化推荐算法研究[D]. 天津：天津师范大学，2015.

第 11 章 基于项目热度的协同过滤推荐算法



本章针对协同过滤推荐算法中采用传统矩阵分解出现负值的现象，提出将NMF和项目“热度”相结合的两阶段 k-NN 近邻选择算法。首先通过 NMF 得到项目的非负隐式特征空间，并使用改进的余弦相似度量项目间相似度，根据 k-NN 算法得到第一阶段 k 近邻集合。接着构建项目“热度”与其相似度相结合的模型，进行第二阶段 k 近邻选择，并完成预测推荐。最后的实验结果表明在数据稀疏情况下，本章提出的 CFNMF-HR(Algorithm of Collaborative Filtering Based on NMF and Heat Range)算法与经典算法相比，鲁棒性更好，推荐精度也得到一定程度的提高。

11.1 引言

在第 4 章提出的改进算法中采用 SVD 矩阵分解时出现负值，此种现象出现的原因是在 SVD 模型中未对用户和项目特征空间进行条件约束。在现实世界中，各个维度中出现负值的向量是不具有现实意义的。例如，在电影推荐系统中，用户隐式向量的某一维度值代表当前用户对当前维度(电影)的喜好程度。相应的，对于一部电影来说，项目隐式向量的某一维度值代表当前电影与当前电影类别的相关程度，若为负值，则无法解释。NMF 可以很好地弥补上述问题，并已广泛应用于协同过滤推荐算法之中。将 NMF(Non negative Matrix Factorization)运用在协同过滤推荐算法中，挖掘出更合理、更符合实际的隐式特征。将非负矩阵应用于对三维数据的分析，将三维数据投影为二维，分别采用非负矩阵分解完成协同过滤过程，有效地提高了预测精度。

针对上述问题，本章提出将非负矩阵分解和项目“热度”相结合的两阶段 k 近邻选择算法。传统的基于项目的协同过滤推荐算法只将项目间的相似度作为唯一因素，而实际现实生活中，“热门”商品对用户购买行为的影响也是一个重要因素。

11.2 非负矩阵分解

给定 $R_{m\times n}$ 评分矩阵，在优化迭代训练过程产生两个非负矩阵 $W_{m\times k}$ 和 $H_{k\times n}$，使得 $R\approx W\cdot H$。即 $R_{m\times n}$ 的列向量是基矩阵 W 中所有列向量的加权和，系数矩阵 H 对应的列向量元素为权重系数且满足 $(m+n)\times k<m\times n$，通过参数 k 值的选取实现系数数据降维。在协同过滤推荐算法中，W 为用户隐式特征向量空间矩阵，H 为项目特征向量空间矩阵，其非负性更具现实意义。

NMF 利用乘性迭代的方法求解 W 和 H，是一个无监督的自学习过程。假设目标函数为 $E\in R_{m\times n}$，易得目标函数的迭代式如式(11-1)所示。

$$E=R-WH \tag{11-1}$$

为了得到的矩阵 W 和 H 使得 $\parallel E\parallel$ 最小，假设噪声服从高斯分布，那么得到最大似然函数如式(11-2)所示。

$$\begin{aligned}L(W,H)&=\prod_{i,j}\frac{1}{\sqrt{2\pi\sigma_{ij}}}\exp\left(-\frac{E_{ij}^2}{2\sigma_{ij}}\right)\\&=\prod_{i,j}\frac{1}{\sqrt{2\pi\sigma_{ij}}}\exp\left\{-\frac{[R_{ij}-(WH)_{ij}]^2}{2\sigma_{ij}}\right\}\end{aligned} \tag{11-2}$$

取对数后，得到对数似然函数为

$$\ln L(W,H)=\sum_{i,j}\ln\frac{1}{\sqrt{2\pi\sigma_{ij}}}-\frac{1}{\sigma_{ij}}\cdot\frac{1}{2}\sum_{i,j}[R_{ij}-(WH)_{ij}]^2 \tag{11-3}$$

假设矩阵各个数据点与真实值间偏差的方差相同，则使得式(11-3)的取值最大，即：①式(11-1)应达到最小值才能实现无限逼近真实值的目的，可得到损失函数；②范数损失函数，其是基于欧几里得距离的度量，如式(11-4)所示。

$$J(W,H)=\frac{1}{2}\sum_{i,j}[R_{ij}-(WH)_{ij}]^2 \tag{11-4}$$

对式(11-5)求导：

$$(WH)_{ij}=\sum_k W_{ik}H_{kj}\Rightarrow\frac{\partial(WH)_{ij}}{\partial W_{ik}}=H_{kj} \tag{11-5}$$

那么得到下式：

$$\begin{aligned}\frac{\partial J(W,H)}{\partial W_{ik}}&=\sum_j[H_{kj}(R_{ij}-(WH)_{ij})]\\&=\sum_j R_{ij}H_{kj}-\sum_j(WH)_{ij}H_{kj}\\&=(RH^{\mathrm{T}})_{ik}-(WHH^{\mathrm{T}})_{ik}\end{aligned} \tag{11-6}$$

同理可得：

$$\frac{\partial J(W,H)}{\partial H_{kj}}=(W^{\mathrm{T}}R)_{kj}-(W^{\mathrm{T}}WH)_{kj} \tag{11-7}$$

优化方法采用梯度下降法进行迭代训练。如式(11-8)、式(11-9)所示。

$$W_{ik}=W_{ik}-\alpha_1\cdot[(RH^{\mathrm{T}})_{ik}-(WHH^{\mathrm{T}})_{ik}] \tag{11-8}$$

$$H_{kj}=H_{kj}-\alpha_2\cdot[(W^{\mathrm{T}}R)_{kj}-(W^{\mathrm{T}}WH)_{kj}] \tag{11-9}$$

式中：$\alpha_1=\dfrac{W_{ik}}{(W^{\mathrm{T}}WH)_{ik}}$；

$\alpha_2=\dfrac{H_{kj}}{(WHH^{\mathrm{T}})_{kj}}$。

最终得到迭代式如式(11-10)、式(11-11)所示。

$$W_{ik}=W_{ik}\cdot\frac{(RH^{\mathrm{T}})_{ik}}{(WHH^{\mathrm{T}})_{ik}} \tag{11-10}$$

$$H_{kj}=H_{kj}\cdot\frac{(W^{\mathrm{T}}R)_{kj}}{(W^{\mathrm{T}}WH)_{kj}} \tag{11-11}$$

式(11-10)和式(11-11)保证迭代结果始终为非负值，反复迭代至目标函数小于等于预设定的阈值为止，完成此优化问题。

11.3 两阶段近邻选择

11.3.1 两阶段 k 近邻选择

本章提出的算法在两个阶段运用 k-NN 算法来寻找项目的最近邻居，第一阶段是在项目非负特征空间下，采用式(4-3)的计算方法得到项目相似度矩阵，在此，对每个项目取 k(k=10，20，30)个邻居，得到项目邻居群 Q_i。第二阶段是在将项目"热度"融入相似度空间后，在项目局部信任空间中再次寻找最近邻居。

11.3.2 项目"热度"和局部信任

单个项目在全部项目中的"热度"由两部分构成，即项目被用户评价的次数和在第一阶段近邻选择时项目被其他项目作为邻居的次数。此概念是用户和项目关联关系的深度挖掘。用式(11-3)表示项目 i"热度"，且 $0\leqslant T_i\leqslant 1$。

局部信任：采用 4.3 节中提出的局部信任的思想。

11.3.3 预测评分

预测用户 u 对项目 i 的评分，得到预测矩阵。计算如式(11-12)所示。

$$p_{u,i}=\frac{\sum_{k=1}^{l}T_i(P_k)*r_{ui}}{\sum_{k=1}^{l}|T_i(P_k)|} \tag{11-12}$$

式中：l——根据 k-NN 算法得到的项目 i 的邻居数。

11.4 算法描述

算法基本思想：首先利用非负矩阵分解得到项目非负特征空间，接着采用式(4-3)的相似度计算方法得到项目间相似度，然后根据 k-NN 算法得到第一阶段

邻居集，在此基础上引入项目的“热度”概念，结合项目相似度得到局部信任；在第二阶段邻居选择，最终完成预测评分。

算法 11-1　CFNMF-HR 算法

输入：评分矩阵 R。
输出：预测矩阵 R_{pred}。

步骤 1：使用非负矩阵分解方法分解 R 得到降维后的矩阵 W 和 H。
步骤 2：在项目非负特征空间下，使用式(2-3)计算项目 i 和项目 j 的相似度 $\mathrm{sim}(i,j)$。
步骤 3：根据 k-NN 算法，设定 $k(k=20)$个邻居，得到项目邻居群 Q_i，由项目邻居群 Q_i 得到 q_i，遍历原始矩阵 R 得到 f_i。
步骤 4：由式(3-3)计算项目的“热度”T_i，并将 T_i 填充为矩阵，使用式(3-4)计算项目间局部信任。
步骤 5：根据 k-NN 算法，得到项目的最近邻居集。
步骤 6：由式(4-12)进行预测评分。
步骤 7：根据 Top-N 方法，将预测评分最高的 N 个项目推荐给相应的用户。
算法结束

CFNMF-HR 算法利用用户和项目之间的潜在关系达到对维空间的简化，使用非负矩阵分解来增加数据的密度，将项目的“热度”作为权重加入到相似度计算中，不仅能克服数据稀疏性问题，还能避免过于强调相似度的作用，鲁棒性更好，提高了推荐的精度。

11.5　实验结果分析

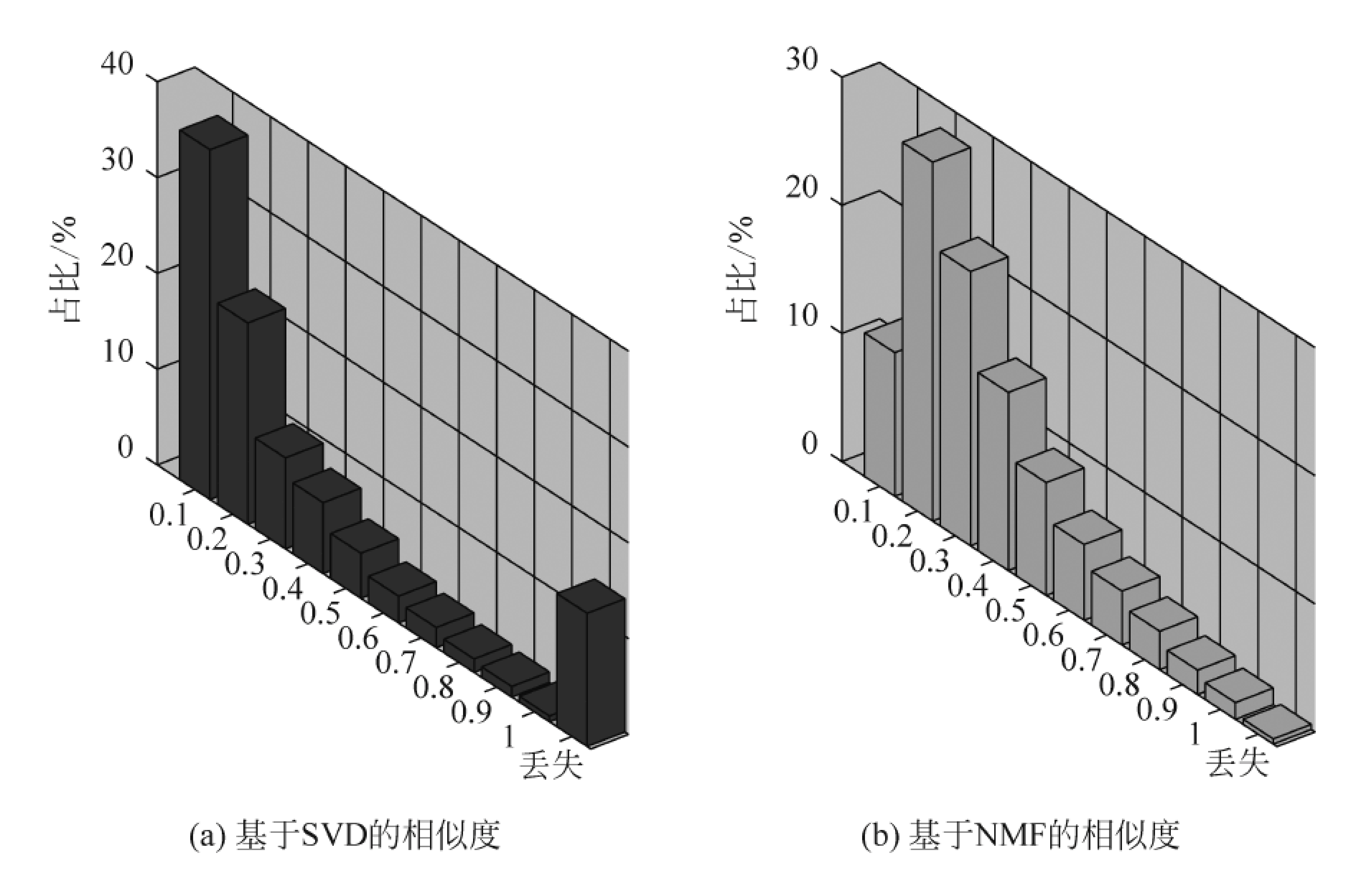
11.5.1　不同策略下相似度的分布

图 11-1 为不同策略下相似度的分布，显示了基于 SVD 和 NMF 下分别得到的项目隐式空间后，采用式(4-3)的相似度计算方法得到的项目间相似度的分布情况。SVD 策略下，相似度值主要分布在[0,0.2]，分布过于极端。NMF 策略下，相似度分布比较均匀，分布在[0,0.5]。

由于 SVD 策略下的隐式空间中存在大量负值，此情况不符合现实，因为项目特征为负值的含义是项目可能为这一类别的概率为负，所以造成了相似度空间中值的丢失，丢失率达 13.9%；而在 NMF 策略下，丢失率仅有 0.4%。通过以上分析，NMF 策略相对于 SVD 策略，具有更好的鲁棒性。

11.5.2　两种因素的分布与分析

图 11-2 为相似度/热度的分布，显示了影响最终预测结果的两个因素(相似度和“热度”)的分布情况。通过实验得出，在第一阶段邻居选择中，邻居数量参数 $k=20$ 时，RMSE 达到最优。以下结果是当 $k=20$ 时的分布情况，相似度主要分布



(a) 基于SVD的相似度　　(b) 基于NMF的相似度

图 11-1　不同策略下相似度的分布

在[0,0.5],“热度”值主要分布在[0.4,0.9]。在推荐系统中,每个项目都可以一个可量化的“热度”指标,符合现实世界中热门商品被热卖的现象。每个项目有多个相似的邻居项目,可协助对未知评分项目进行合理预测评分。因此,相似度和“热度”是完全不同的两个因素,把项目的“热度”概念引入协同过滤推荐算法是可行的。

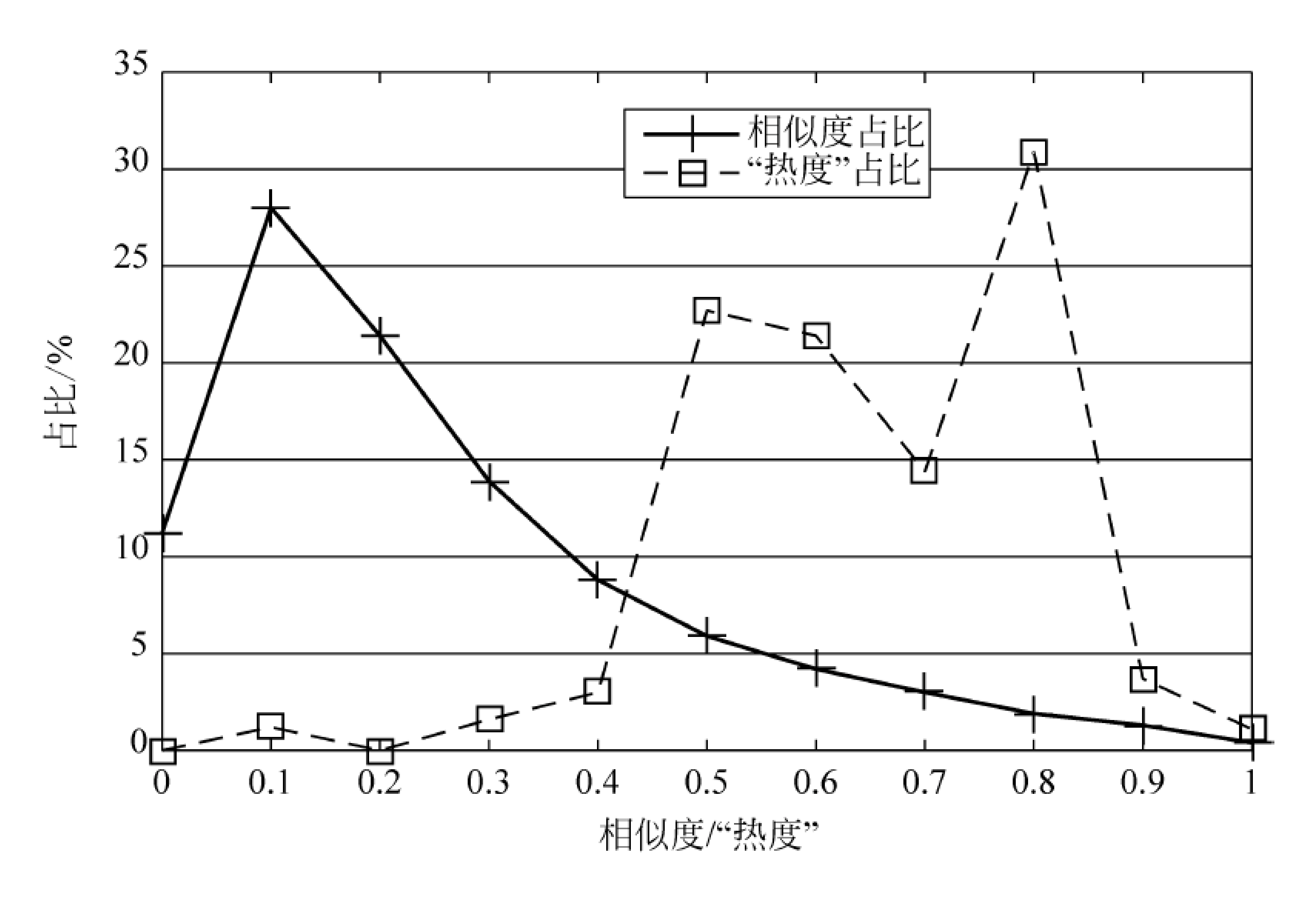


图 11-2　相似度/“热度”的分布

11.5.3　实验结果及分析

图 11-3 为不同推荐策略的 RMSE。3 种算法中 RMSE(参考 4.4.1 节)值随着邻居数目的变化而发生变化,相比传统的基于项目的协同过滤 ICF 算法和传统的

基于 SVD 的协同过滤 SVD-CF 算法，本章提出的 CFNMF-HR 算法在邻居数小于等于 10 个时，RMSE 值呈现指数式的下降；当邻居数大于 10 个时，变化趋稳；当邻居数等于 14 时，推荐性能达到最好，RMSE＝0.9651，比 SVD-CF 精度提高了 1.67％。随后随着邻居数目的增加 RMSE 又开始反弹，说明邻居数目对于算法的影响较大。CFNMF-HR 算法和 SVD-CF 算法相对于 ICF 算法，RMSE 优化更明显。CFNMF-HR 算法又相比 SVD-CF 算法，RMSE 值一直处于下降状态并且推荐精度更优，说明本章提出的算法是有效可行的，具有更好的鲁棒性，不仅有效地增大数据密度，还提高了协同过滤推荐算法的预测精度。

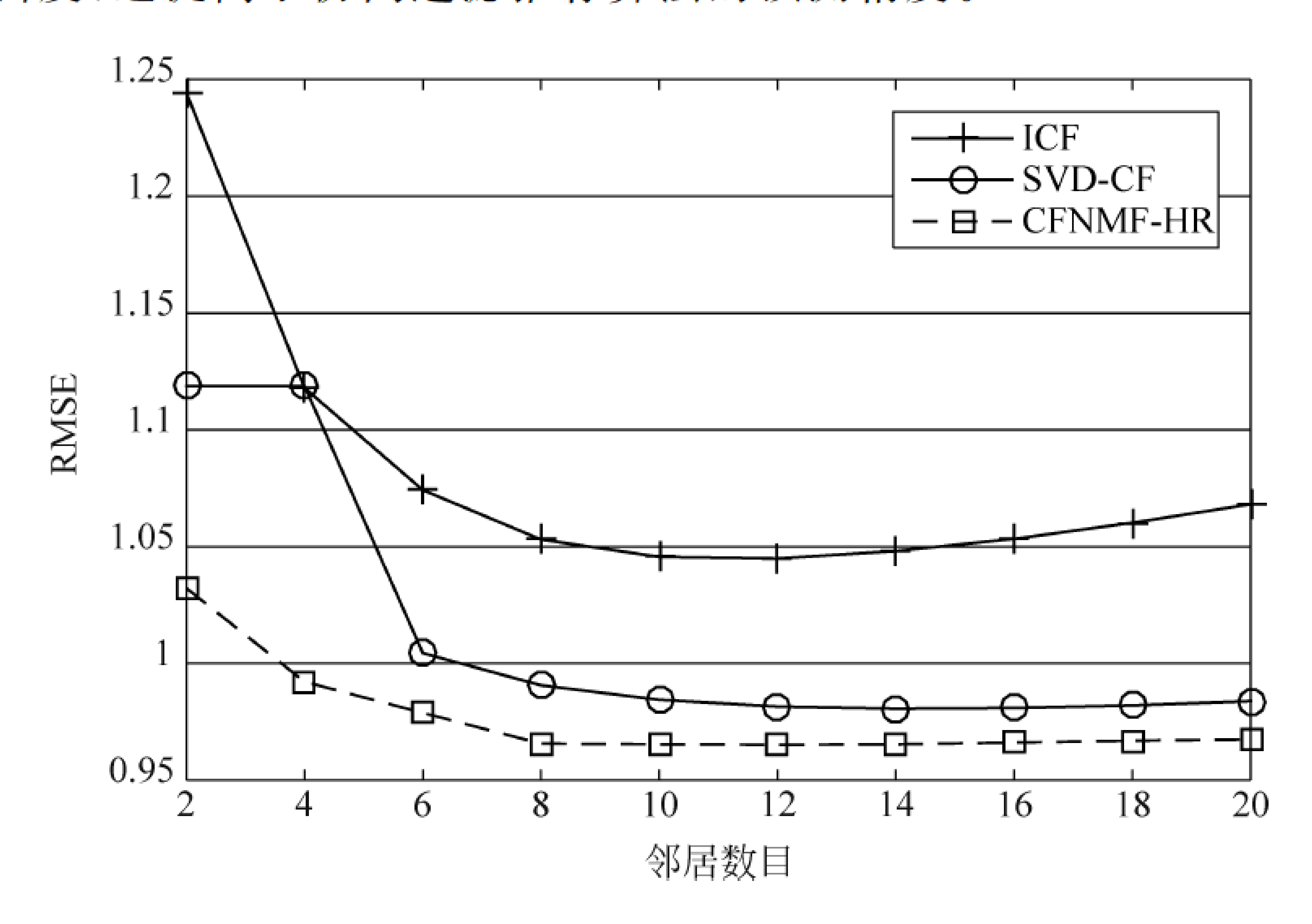


图 11-3　不同推荐策略的 RMSE

本章小结

本章提出将非负矩阵分解 NMF 和“热度”模型相结合的协同过滤推荐算法。首先运用 NMF 得到项目的非负隐式特征空间。然后用改进的余弦相似度计算项目间相似度，生成第一阶段邻居集。接着引入项目“热度”，构建项目“热度”模型并融入到相似度模型中，以做出更高精度推荐预测。最后在 MovieLens 数据集对本章提出的算法进行验证。该算法弥补了 SVD 出现负值的问题，并且增强了算法的鲁棒性，提高了算法的预测精度。

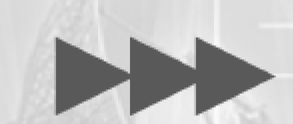
参考文献

[1]　Yin F. Sparsity-Tolerated Algorithm with Missing Value Recovering in User-based Collaborative Filtering Recommendation[J]. Journal of Information & Computational Science，2014，10(15)：4939-4948.

[2] Krzywicki A, Wobcke W, Kim Y S, et al. Collaborative Filtering for people-to-people recommendation in online dating: Data analysis and user trial[J]. International Journal of Human-Computer Studies,2015,76(C): 50-66.
[3] Shahmohammadi A, Khadangi E, Bagheri A. Presenting new collaborative link prediction methods for activity recommendation in Facebook [J]. Neurocomputing, 2016, 210: 217-226.
[4] Zhou X, He J, Huang G, et al. SVD-based incremental approaches for recommender systems [J]. Journal of Computer & System Sciences,2015,81(4): 717-733.
[5] Konstan J A, Miller B N, Maltz D, et al. GroupLens: applying collaborative filtering to Usenet news[J]. Communications of the Acm,1997,40(3): 77-87.
[6] 王稳寅. 针对冷启动推荐的分布式协同过滤研究[D]. 上海: 上海交通大学,2012.
[7] 周子亮. 结合非负矩阵分解的推荐算法及框架研究[D]. 北京: 北京交通大学,2012.
[8] 张明敏. 基于 Spark 平台的协同过滤推荐算法的研究与实现[D]. 南京: 南京理工大学,2015.
[9] 党国健. 基于双重模糊聚类的协同过滤推荐算法研究[D]. 西安: 西安理工大学,2011.
[10] 林丽金,李文翔,杨俊贤,等. 基于协同过滤在高校学习资源个性化推荐系统中应用研究[J]. 价值工程,2016,35(4): 191-193.
[11] Deerwester S. Indexing by latent semantic analysis[J]. Journal of the Association for Information Science and Technology,1990,41(6): 391-407.
[12] Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems [J]. Computer,2009,42(8): 30-37.
[13] Stark C. Top-*N* recommendations from expressive recommender systems[J]. Computer Science,2015.
[14] 许海玲,吴潇,李晓东,等. 互联网推荐系统比较研究[J]. 软件学报. 2009,20(2): 350-362.
[15] Felfernig A, Jeran M, Ninaus G, et al. Toward the Next Generation of Recommender Systems: Applications and Research Challenges [M]. IEEE Educational Activities Department,2005: 734-749.
[16] 孙艳,朱玉全,陈耿. 基于隐语义模型的协同过滤图书推荐方法[J]. 信息技术. 2015(11): 41-44.
[17] Guo G, Zhang J, Yorke-Smith N. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings[C]//2015.
[18] 李远博,曹菡. 基于 PCA 降维的协同过滤推荐算法[J]. 计算机技术与发展,2016(02): 26-30.
[19] Amatriain X, Lathia N, Pujol J M, et al. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web[C]//2009.
[20] Hwang W S, Lee H J, Kim S W, et al. Efficient recommendation methods using category experts for a large dataset[J]. Information Fusion,2016,28(C): 75-82.
[21] Sang H C, Cho Y H. An utility range-based similar product recommendation algorithm for collaborative companies[J]. Expert Systems with Applications,2004,27(4): 549-557.
[22] Hwang W S, Lee H J, Kim S W, et al. Efficient recommendation methods using category experts for a large dataset[J]. Information Fusion,2016,28(C): 75-82.

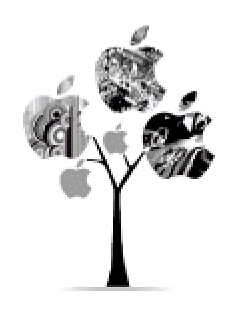
[23] Cho J, Kwon K, Park Y. Collaborative Filtering Using Dual Information Sources[J]. IEEE Intelligent Systems, 2007, 22(3): 30-38.
[24] Jiang W, Yang L. Research of improved recommendation algorithm based on collaborative filtering and content prediction[C]//2016.
[25] Wang Q M, Liu X, Zhu R, et al. A New Personalized Recommendation Algorithm of Combining Content-based and Collaborative Filters[J]. Computer & Modernization, 2013, 1(8): 64-67.
[26] Goldberg D, Oki B M, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the Acm, 1992, 35(12): 61-70.
[27] Resnick, Paul, Iacovou, et al. GroupLens: an open architecture for collaborative filtering of netnews[J]. Working Paper, 1994: 175-186.
[28] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//2001.
[29] Bobadilla J, Ortega F, Hernando A, et al. A collaborative filtering approach to mitigate the new user cold start problem[J]. Knowledge-Based Systems, 2012, 26: 225-238.
[30] Hofmann T. Latent semantic models for collaborative filtering[J]. Acm Transactions on Information Systems, 2013, 22(1): 89-115.

第四篇　基于信任的协同过滤推荐算法



传统的推荐个性化推荐算法,推荐的往往都是热门的商品,这样造成热门的商品更加热门,而处在“长尾分布”上的商品得不到重视。尤其是在数据稀疏性情况下,为提高数据利用率,研究人员将用户间的社会关系信息融入到推荐系统中,提出了基于信任的协同过滤推荐算法,达到优化推荐精度的作用。

第 12 章 带偏置的专家信任推荐算法



本章针对传统基于用户的协同过滤推荐算法较少考虑信任对象所处环境的实时变化,评价项目较少用户的评分预测准确度不高的问题,提出一种带偏置的专家信任推荐算法。为使对象之间的信任度得到较好的量化,合理预测用户评分,首先利用专家的评价可信度、活跃度、评价偏差度计算得到专家的信任值。其次在形成评分的过程中把改进专家算法与用户、项目偏置算法相融合,充分利用偏置信息,以便获得更加客观的预测评分。最后实验结果表明,在不同数据集上相比于传统的推荐算法,该算法在评价项目较少用户的预测准确度方面有显著提高。

12.1 引言

推荐系统的基本功能是利用用户对项目的历史评价信息产生推荐。大数据时代用户接触的信息呈指数级,实际操作过程中,由于用户精力有限,造成大量项目未得到评分,导致评价矩阵极为稀疏。传统协同过滤算法的稀疏用户(评价项目相对较少的用户)评分预测准确度较低,如何提高稀疏用户的推荐准确性成为比较热门的研究课题。

针对稀疏用户评分预测问题,常采用的处理方法有:

(1) 数据平滑算法:对用户尚未评分的项目进行填充。文献[6]运用其他用户对该项目的评分均值填充缺省值;文献[7]将用户信息聚类后,以分类用户的评分来填充缺省值。

(2) 专家算法:把项目按类别划分,找到每个类别的专家,利用专家评分预测用户对项目评分。文献[8]提出了把专家算法与传统协同过滤技术相结合;文献[9]提出了“明星用户”的算法,在评分预测阶段加权平均所有“明星用户”的评分。

以上两类算法一定程度上缓解了稀疏用户评分预测准确性不高的问题,但是计算得到稀疏用户与用户、稀疏用户与各个“专家”之间的相似度区分不明显,无法合理利用用户与专家的预测信息。鉴于此,本章提出 IBETA 算法(Improved With Biased Expert Trust Algorithm,IBETA),该算法在专家算法的基础上添加专家信

任，在评分形成的过程中区别对待不同级别专家的预测值，同时将独立于用户—项目评分以外的因素添加到评分预测公式中，使预测值更加合理。

12.2 相关工作

12.2.1 专家算法

定义 12.1 对于 A 类项目，专家 E_c 的定义如式(12-1)所示。

$$|I_u| \leqslant |I_v| \quad (\forall u \in U - E_c, \forall v \in E_c) \tag{12-1}$$

式中：I_u——用户 u 评价所有项目的集合；

I_v——专家 v 评价的项目集合；

U——所有用户集合。

统计每个用户评价 A 类项目数量，当用户评价项目的数量使式(12-1)成立时，该用户被定义为“专家”。

12.2.2 生成推荐值

Cho 提出的专家算法在计算预测评分时，采用无条件相信专家的策略(Expert Algorithm，EA)，评分预测采用式(12-2)所示。

$$p_{u,i,c} = \bar{r}_{u,c} + \frac{1}{k}\sum_{v \in E_c}(r_{v,i} - \bar{r}_{v,c}) \tag{12-2}$$

式中：$p_{u,i,c}$——当前用户对属于 c 类电影 i 的预测评分；

$\bar{r}_{u,c}$——用户 u 对 c 类电影评分的平均值；

$r_{v,i}$——专家 v 对项目 i 的评分；

$\bar{r}_{v,c}$——专家 v 对 c 类项目评分的平均值；

E_c——c 类项目专家集合。

需要预测的项目属于多个类别时，采用式(12-3)计算评分值。

$$p_{u,i} = \frac{1}{|c_i|}\sum_{c \in c_i} p_{u,i,c} \tag{12-3}$$

式中：c_i——项目所属的类数；

$p_{u,i,c}$——每一类预测的值。

以上预测评分算法的运行时间复杂度较低，但稀疏用户评分预测准确性较低，原因是该算法在项目确定的情况下，对于不同用户的预测分数几乎是一样的，因为专家的选择只考虑了当前用户需要预测的项目；在选定专家集合中，同等对待每个专家的预测值。专家的专业水平有高有低，以上算法显然有失偏颇。

Breeze 提出“专家与相似度结合算法”(Expert Similarity Algorithm，ESA)在计算预测评分时，采用把专家看成某种意义上的近邻，在预测评分时根据专家与当前用户相似度的大小赋予不同的权重值，评分预测如式(12-4)所示。

$$p_{u,i,c} = \bar{r}_{u,c} + \frac{\sum_{v \in E_c} (r_{v,i} - \bar{r}_{v,c}) \cdot s_{u,v}}{\sum_{v \in E_c} s_{u,v}} \tag{12-4}$$

式中：$s_{u,v}$——专家与当前用户的相似度。

与 EA 算法相比，ESA 算法对于不同的专家赋予了不同的权重值。

随着稀疏性的增加，稀疏用户与各个专家的邻近程度区分不明显，计算量增加的同时稀疏用户的推荐准确性并没有实质性的提高。

12.2.3 Baseline 预测

评估一个策略的性能好坏，需要建立一个对比基线，在对比基线的基础上观察后续实验效果的变化。观测到的评分数据有一些和用户无关的因素产生的效果，即一部分因素是和用户对物品的喜好无关而只取决于用户或物品本身的特性。例如，乐观积极的用户对于一些项目的评分普遍较高，而悲观消极的用户对项目的评分普遍较低，也就是说即使这两类用户对同一项目的评分相同，但是对物品的喜好程度确是不一样的。对于项目来说道理是一样的，受用户欢迎的项目评分普遍较高，不受用户欢迎的项目评分较低，加入偏执信息的评分预测公式如式(12-5)表示。

$$R^*(i,j) = \mu + b(i) + b(j) \tag{12-5}$$

式中：$R^*(i,j)$——用户 i 对项目 j 的预测评分；

μ——数据集的总体偏置信息；

$b(i)$——用户 i 的偏置信息；

$b(j)$——项目 j 的偏置信息。

假设项目的总体偏置为 a，项目 1 的口碑普遍高于其他项目的值为 b，如果 u_1 是悲观严谨的，其 $b_u(i)$ 值为 c，那么根据式(12-5)u_1 对项目 1 的预测值为 $a+b-c$。

计算 $b(i)$ 和 $b(j)$ 的值，采用式(12-6)、式(12-7)求解。

$$b(i) = \frac{\sum_{i \in I} R(i,j) - \mu - b(i)}{\lambda_1 + |I|} \tag{12-6}$$

$$b(j) = \frac{\sum_{j \in U_i} R(i,j) - \mu}{\lambda_2 + |U_i|} \tag{12-7}$$

式中：i——用户；

j——项目；

I——用户 i 评价过的项目集合；

μ——数据集的总体偏置信息；

$|I|$——集合的个数；

U_i——评价过项目 j 的用户集合；

$|U_i|$——集合的个数；

参数 λ_1,λ_2 需要实验确定。

12.3 改进专家算法

从专家算法提出至今，许多研究人员都围绕着利用专家与用户、项目的关系提升推荐准确度。但是现实生活中人们在参考权威人士的意见时，必然要考虑权威人士的可信度。到目前为止推荐系统并没有对“信任”给出一个具体的概念。在可查资料中，信任是指接受推荐者对提供推荐者特定行为的主观可能性预测。在社交网络中信任需要考虑的因素有很多，完整地考虑各个方面难度很大且通常没有太大必要，在面对同一个用户时，只需要对该用户所处的情形进行相应的加强和减弱，以便于对对象之间的信任程度进行较好的量化。

12.3.1 改进专家信任

在推荐系统中存在着各种各样的数据，其中包括了评分数据、项目属性数据、用户属性数据等，这些数据基本构成了本章需要的信任度量情境。充分考虑专家及用户所在环境，本章用以下定义量化专家信任中涉及的重要概念。

定义 12.2 专家评价可信度

一个专家评价的项目数量越多，可以从一定程度上反映出其评价项目的质量、可信度，度量专家评价可信度可以用式(12-8)表示。

$$D_u = \frac{Q_u}{\max(Q_{\text{all}})} \tag{12-8}$$

式中：Q_{all}——指所有用户；

Q_u——专家 u 评价过的所有项目的集合；

$\max(Q_{\text{all}})$——所有专家中评价项目的最大值。

定义 12.3 专家专业度

咨询专家意见之前，人们通常会考虑专家的专业度。专家并不是对所有种类的项目都具有全面的专业知识，在某种情况下，一名专家显然只会对一个或者很少种类的项目上投入比较多的精力，具体表现为在某一类项目上评价比较多的项目，因此专家专业度用式(12-9)表示。

$$R_u = \frac{T_{ui}}{T} \tag{12-9}$$

式中：T_{ui}——专家已评价且属于某一种类的所有项目集合；

T——系统中获得过用户评价且属于这一主题的所有项目集合。

定义 12.4 专家评价偏差度

专家计算的预测评分与真实评分之间的差值为专家评价偏差度，如式(12-10)

所示。

$$P_u = \frac{Z_u}{Q_u} \tag{12-10}$$

式中：Z_u——最小偏差项目的集合。

在计算 Z_u 时，利用项目评分的平均值 δ 表示项目的真实质量，专家评价的偏差如果小于 δ，则把此评分项目加入到 Z_u 中，对专家 u 及项目 i：如果 $|r_{u,i}-\overline{r_i}|\leqslant\delta$ 成立，则 $i\in\delta$；在本实验中 δ 如果太小则 Z_u 趋于零，计算就没有太大意义；太大会使 δ 趋于 1，计算效果不理想。本实验 δ 取为 0.37。

基于以上表述专家的 D_u，P_u，R_u，权重系数 w_1，w_2，w_3 及式(12-11)计算专家信任值。

$$TR_u = w_1 \cdot D_u + w_2 \cdot P_u + w_3 \cdot R_u \tag{12-11}$$

式(12-11)中权重系数需要采用原始专家算法 CE 矫正，调节原理是设置一个初始 RMSE 值，利用评分预测公式计算预测值并计算一次 RMSE 值，当前 RMSE 值大于设定 RMSE 值时，利用控制变量法更新一次权重系数，记录每个参数达到最优 RMSE 值时的值，最后归一化处理三个参数得到最终权重系数值。

12.3.2　评分形成

由于传统相似度计算对稀疏用户基本失效，本章把专家可信度、用户、项目偏置与专家算法相结合，如式(12-12)所示生成预测值。

$$p_{u,i} = \frac{1}{\sum\limits_{c\in C_i} f_{u,c}} \times \sum_{c\in C_i}\left(\bar{r}_{u,c} + \frac{\sum\limits_{v\in E_c}(r_{v,i}-\bar{r}_{v,c})\cdot TR_v}{\sum\limits_{v\in E_c} TR_v}\right)\cdot f_{u,c} + bu(i) + bi(j) \tag{12-12}$$

式中：TR_v——专家可信度；

C_i——当前项目所属类别总数；

$f_{u,c}$——用户 u 评价的 c 类项目占所有 c 类项目的百分比；

$bu(i)$——用户 U 的偏置；

$bi(j)$——项目 i 的偏置。

该算法对于每个电影类别的专家评分，根据专家在此类别评价项目中的信任值，加权计算预测评分，有效避免了不同类别专家对项目的评分同等对待的问题，根据专家信任值赋予不同专家不同的权重，在一定程度降低了预测误差。确定专家依据的是用户的历史评价信息，在该过程中主观因素起决定性作用，为了提升算法的鲁棒性，在形成评分的过程中需要考虑独立于评分以外的客观因素。本章把用户、项目偏置引入到评分预测公式中，在改进专家算法形成评分的基础上引入用户、项目偏置，进一步提升了预测准确性及合理性。

12.3.3 算法描述

算法 12-1 CFNMF-HR 算法

输入：评分矩阵 R 及项目类别矩阵 T，RMSE 阈值 0.98，Round 值。
输出：预测矩阵 R_{pred}。

步骤 1：数据预处理；Baseline 预测确定数据集上 λ_1，λ_2 的值；初始化 w_1，w_2，w_3。
步骤 2：利用定义 12.1 确定每个类别的专家。
步骤 3：根据评分偏差修正一次 w_1，w_2，w_3，直到出现最优 RMSE 值。

```
for i=1：round
  根据式(12-2)计算预测值并计算 RMSE 值。
  if(当前 RMSE 值>设定 RMSE 值)
    修正一次 w1，w2，w3 的值(控制变量法)；
  end
  i++；
end
```

步骤 4：根据式(12-6)、式(12-7)计算用户、项目偏置。
步骤 5：根据步骤 2 寻找到的专家及式(12-12)，形成预测值。
步骤 6：产生预测矩阵 R_{pred}。
算法结束

12.4 实验结果与分析

12.4.1 数据集

本实验分别在 Epinion、Ciao、MovieLens 三个数据集进行，这三个数据集都包含了用户对项目的评分且分值为 1～5 的离散值，数据集的具体信息如表 12-1 所示。

表 12-1 实验数据集

数据集	用户数目	项目数目	评分数
MovieLens	943	1682	100 000
Ciao	7375	106 797	284 086
Epinion	22 166	296 277	922 267

12.4.2 评估标准

评估推荐系统预测准确性的标准分为决策精度标准和统计精度标准两类。本章采取了对特大或特小误差反应敏感的均方根误差(RMSE)。在推荐系统中 RMSE 作为一种常用度量误差标准被广泛使用，其原理是通过计算用户关于项目的

预测值与真实值之间的偏差平方和与用户个数 n 比值的平方根，如式(12-13)所示。

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i}^{n}(x_i-x_o)^2}{n}} \tag{12-13}$$

式中：x_i——预测值；

x_o——与预测值对应的真实值。

12.4.3　实验结果及分析

1. Baseline 预测

将数据集的 90%作为训练集，其余的 10%作为测试集。首先根据项目评分的平均值确定为数据集的总体偏置 μ，其次根据式(12-6)、式(12-7)及初始化的 λ_1，λ_2 计算用户及项目的偏置，调整 λ_1，λ_2 的值提高 Baseline 的预测 RMSE 值。选择 Baseline 预测的目的在于该算法的训练时间短，预测精度高，可以通过实验训练得到最优参数。

如图 12-1 所示是 MovieLens 数据集的 Baseline 预测。经实验发现当 $\lambda_1=3$，$\lambda_2=6$，RMSE 达到最优，最小值为 0.964。

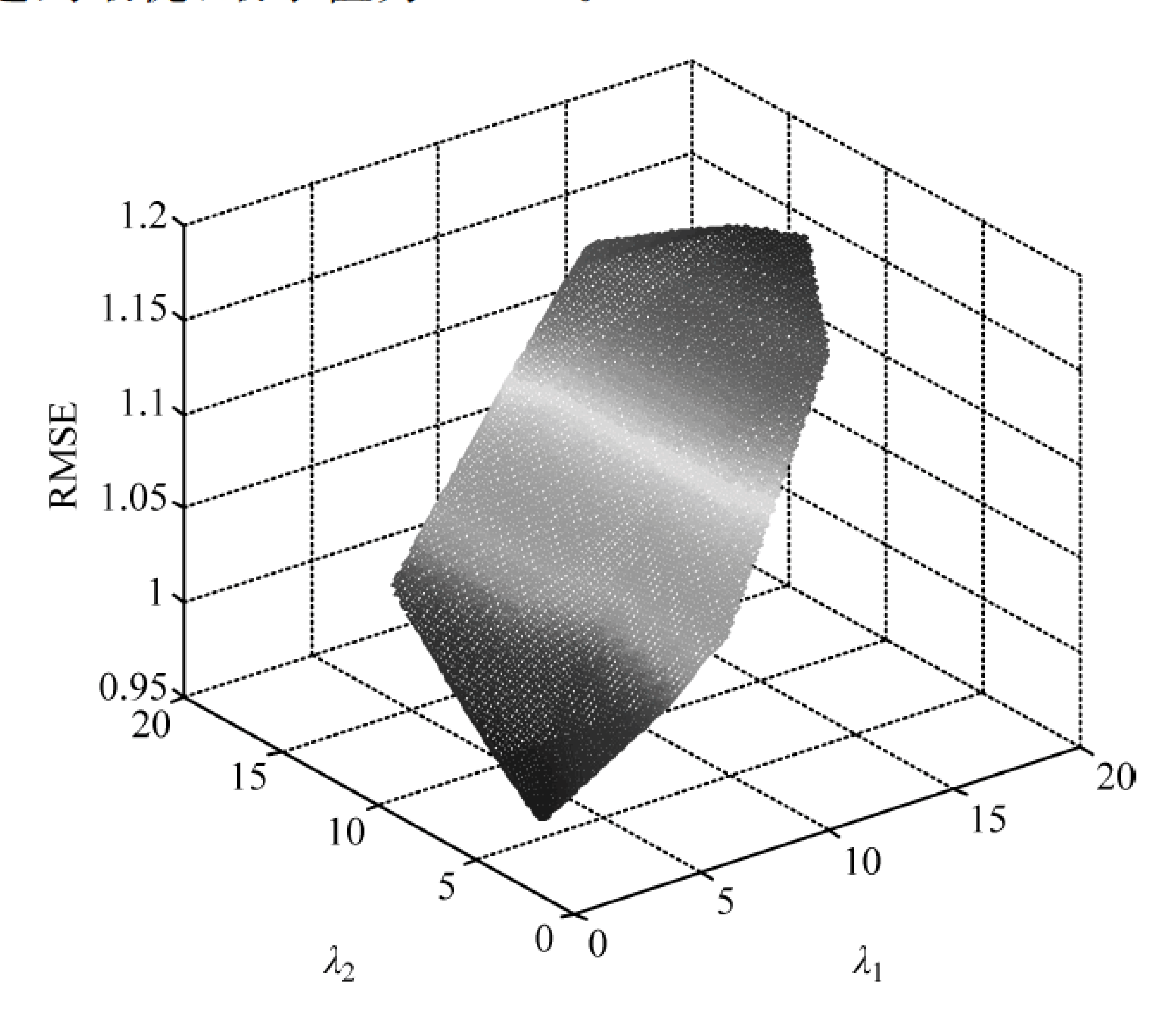


图 12-1　MovieLens 数据集 Baseline 预测

如图 12-2 所示是在 Ciao 数据集的 Baseline 预测。经实验发现当 $\lambda_1=58$，$\lambda_2=43$，RMSE 达到最优，最小值为 0.976。

如图 12-3 所示是在 Epinions 数据的 Baseline 预测。经实验发现当 $\lambda_1=53$，$\lambda_2=56$，RMSE 达到最优，最小值为 0.998。

2. 用户可信度指标分布与分析

以 MovieLens 1M 数据集为例，图 12-4 为评价指标分布图，给出用户专业度、评价可信度、评价偏差度的分布情况。其中 58.21%的用户专业度分布在[0,0.2]，

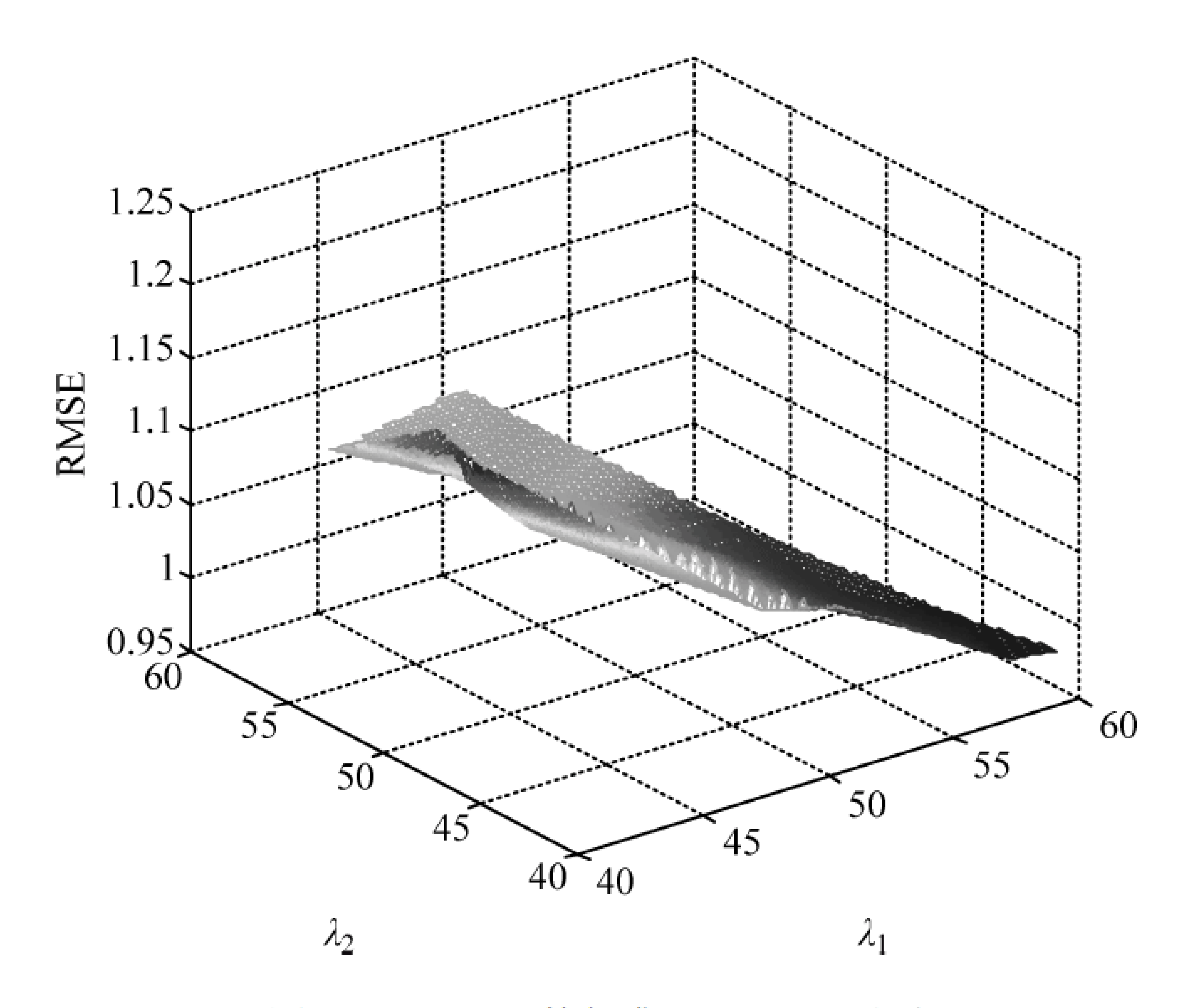


图 12-2　Ciao 数据集 Baseline 预测

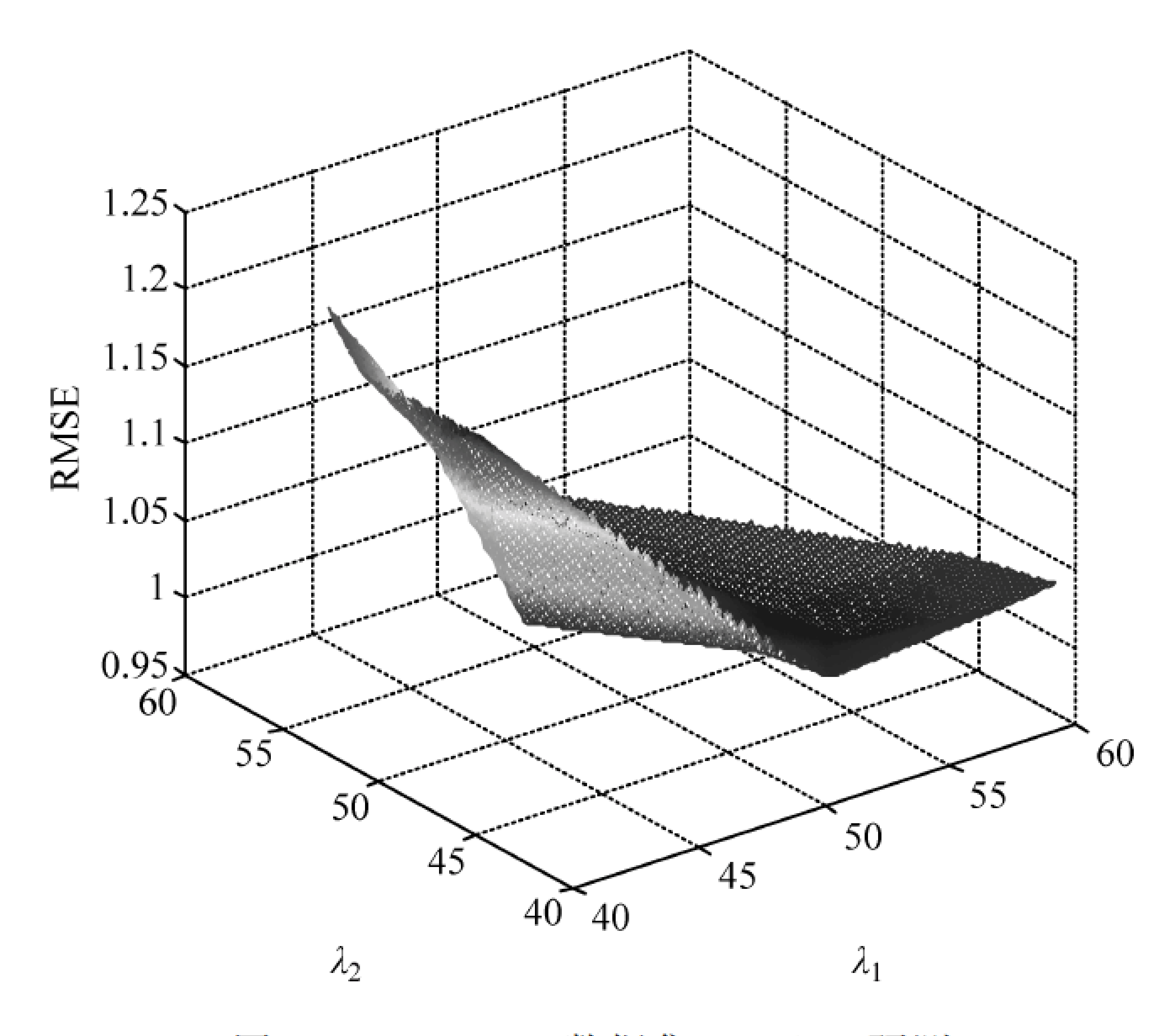


图 12-3　Epinion 数据集 Baseline 预测

56.4%的用户评价可信度分布在[0,0.1],说明多数用户的专业度、评价可信度两种指标较低,少数用户评价可信度能在全体的用户评价可信度中体现个性化的特质。从图 12-4 可以看出,用户的评价偏差度分布几乎呈正态分布,评价偏差度分布在[0.2,0.6]的用户所占比例为 77.6%,说明大多数用户的评价偏差度就比较高(评价比较接近真实评分)。以上足以说明选定专业用户以后(专家),该专家对项目的评分信任度可以由以上三种指标体现。

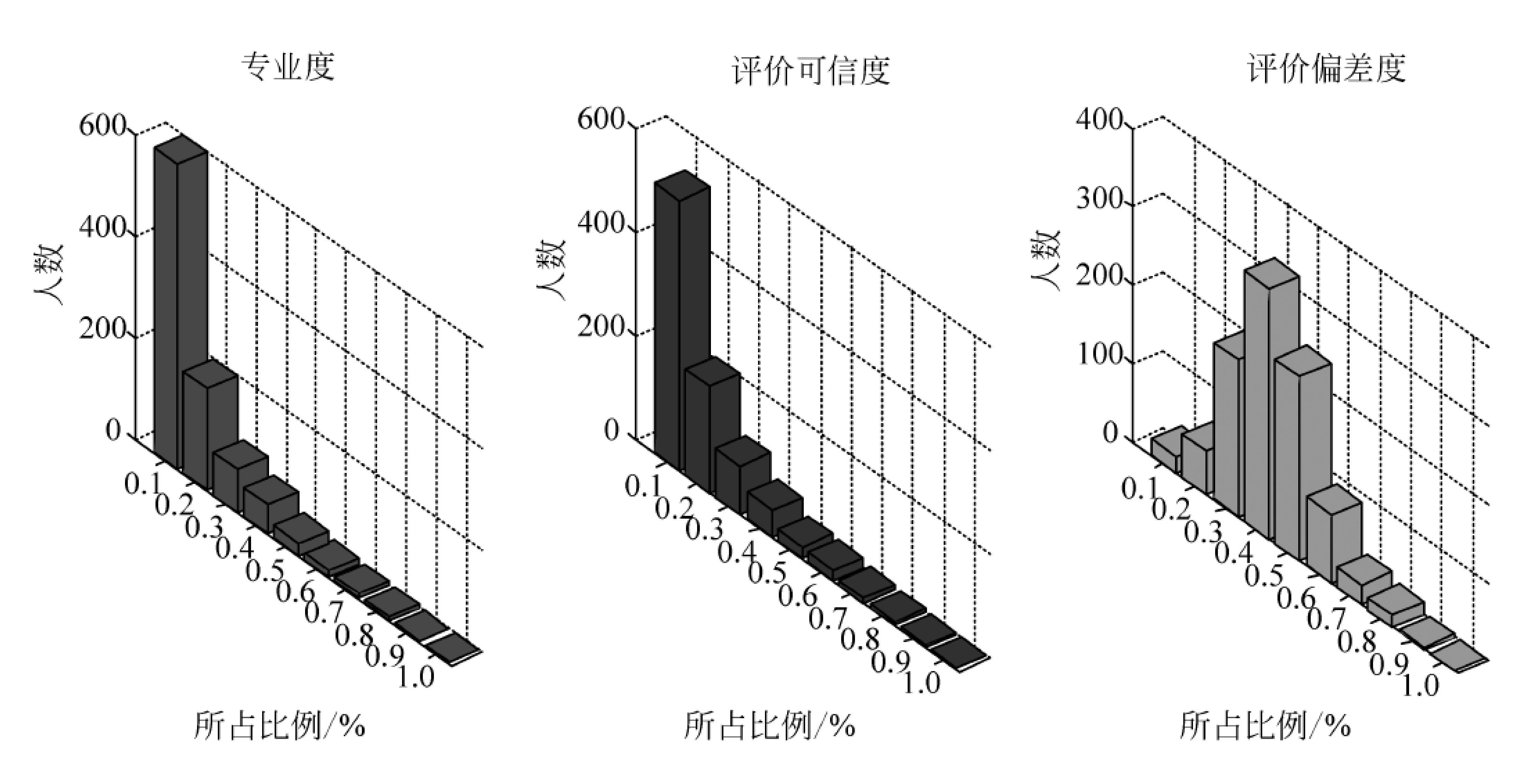


图 12-4　评价指标分布图

3. 稀疏用户分布

对于大型数据集，稀疏用户的数量比较大。图 12-5 为用户评价项目数量分布图，按照用户评价项目数量，把 MovieLens 1M 数据集、Ciao 数据集、Epinion 数据集的用户分为四类，可以看出三种数据集第一类用户（本章视为稀疏用户）占全部用户的比例较大。

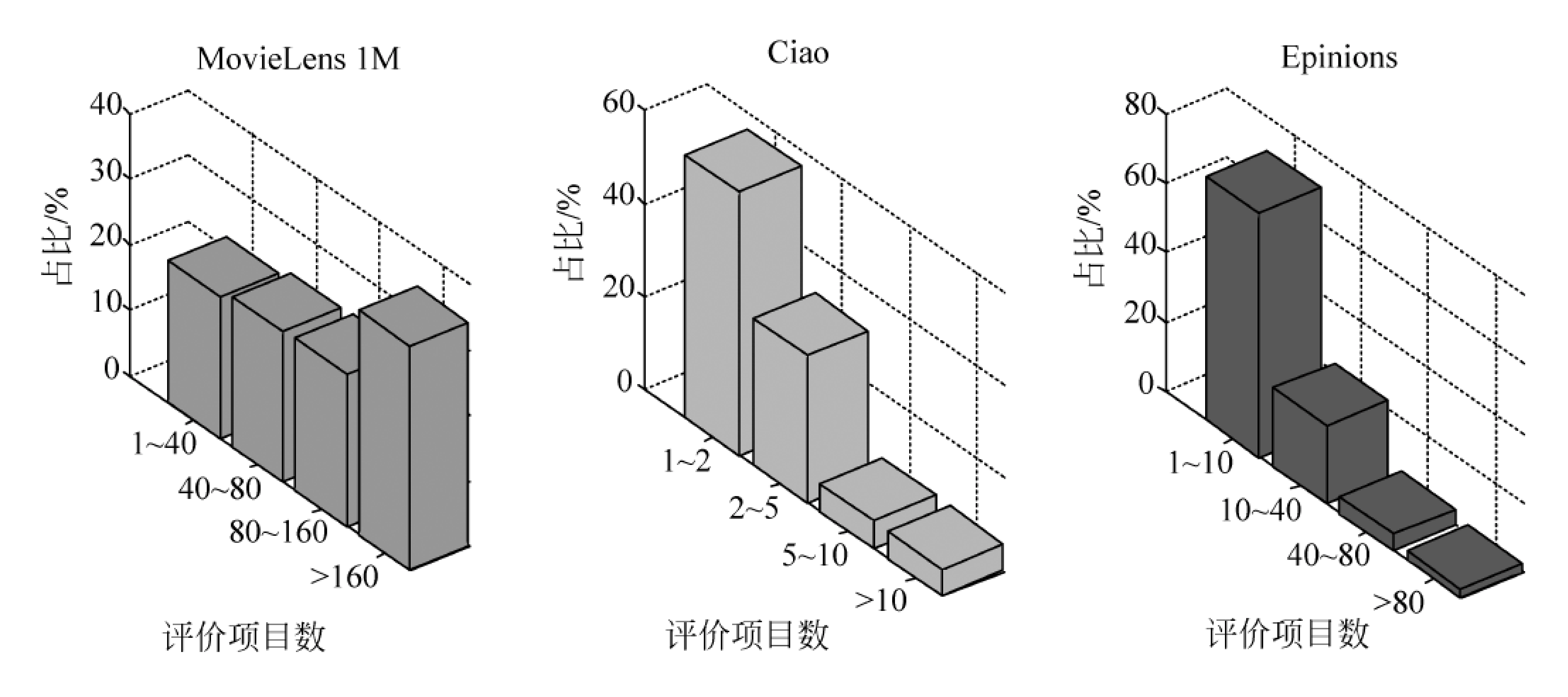


图 12-5　用户评价项目数量分布图

4. 专家可信度分布与分析

以 MovieLens 1M 数据集为例，图 12-6 为专家可信度分布图，表示专家信任值随着矫正次数的变化情况。在实验开始阶段，专家可信度指标之间系数是初始化值，所以专家可信度比较低，随着矫正次数的增加专家信任值从 100 次矫正的 0.12 提升到 0.63 共经过了 400 次的矫正，从 400 次开始专家信任值稳定在[0.63, 0.69]，此时得出 w_1，w_2，w_3 值分别为 0.31，0.46，0.23，在后续计算专家对此类别

项目评分时,可以直接利用此训练后的系数值,但是对于不同类别的专家信任因子系数值不同,需要同样的方法训练得到。

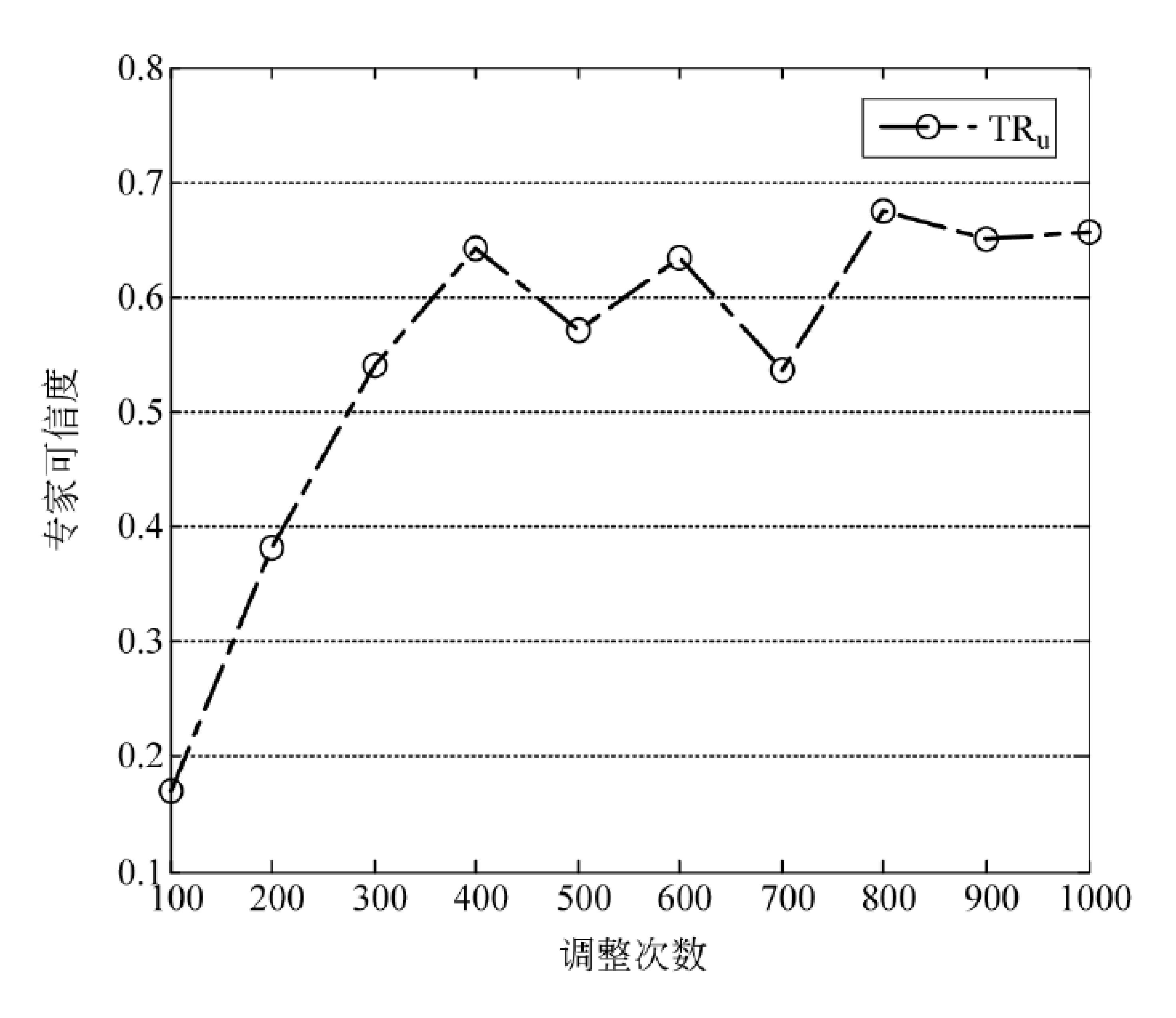


图 12-6　专家可信度分布图

5. 在不同数据集上稀疏用户预测对比

如图 12-7 至图 12-9 所示分别为 MovieLens 1M 数据集、Ciao 数据集和 Epinion 数据集上三种算法对比。可以明显看出,在实验开始阶段三种算法针对稀疏用户的预测准确度较低。随着专家人数的增加,相比于传统的 EA、ESA 算法,本章提出的 IBETA 算法在三种数据集下 RMSE 有普遍提高,原因是随着独立于用户与项目的偏置信息的加入,预测稀疏用户的项目评分增加了客观的预测值。

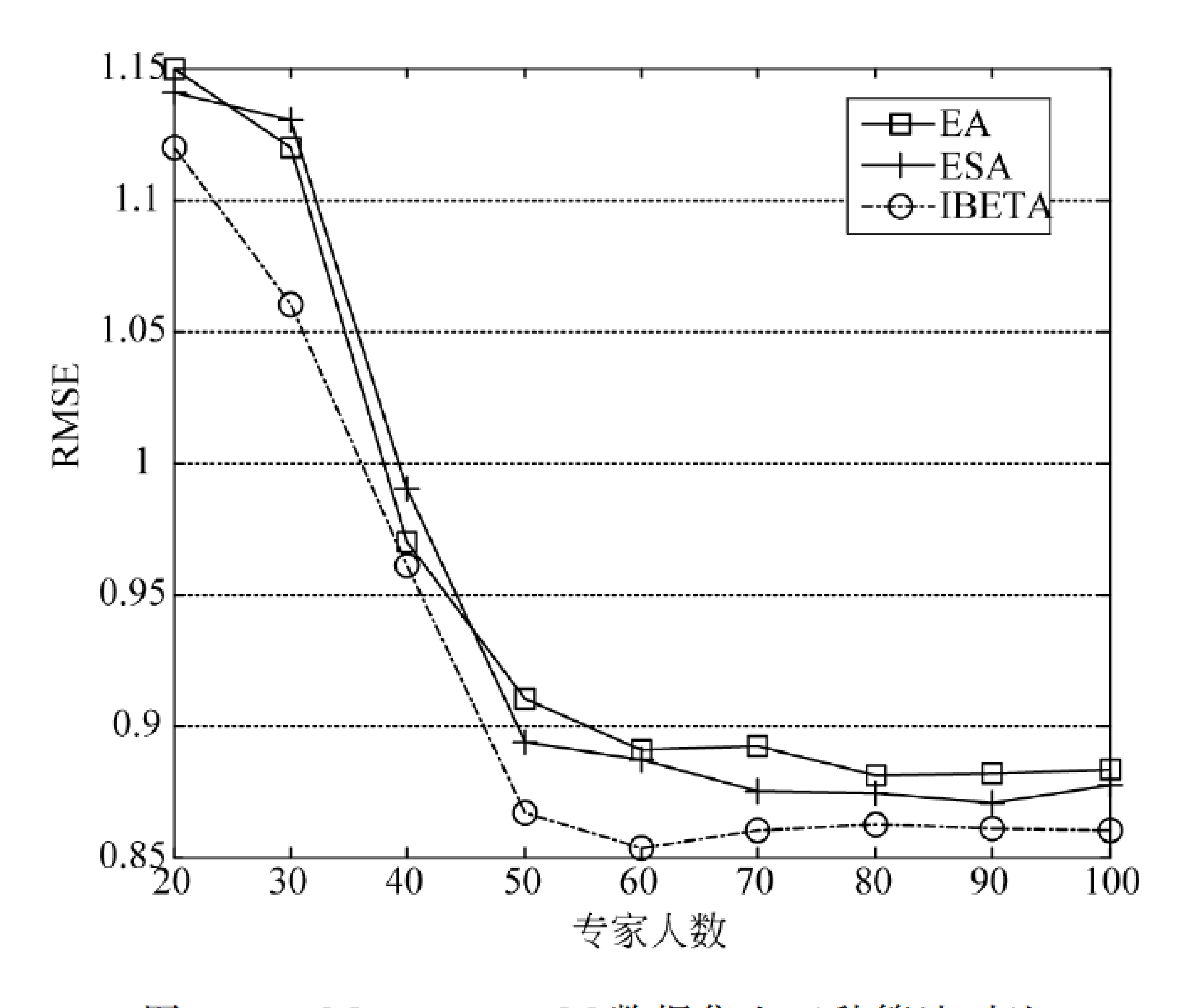


图 12-7　MovieLens 1M 数据集上三种算法对比

随着稀疏用户评价项目的增多，三种算法的预测准确性都有明显的上升趋势，原因是随着用户评价项目的增多，相似度计算能够更加明确地区分出用户与专家之间的相似度，由近邻算法的特点不难理解预测准确性的提升。

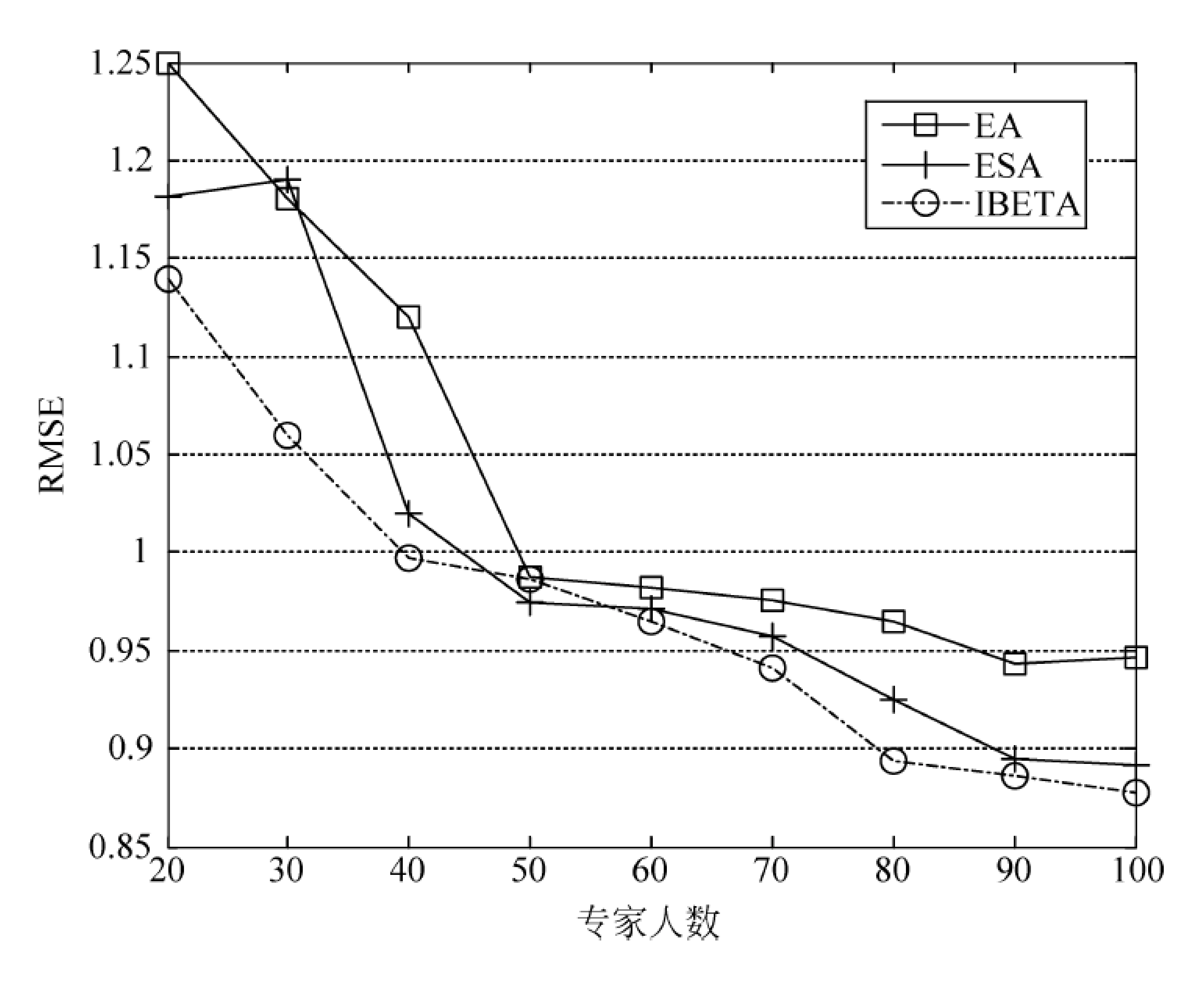


图 12-8　Ciao 数据集上三种算法对比

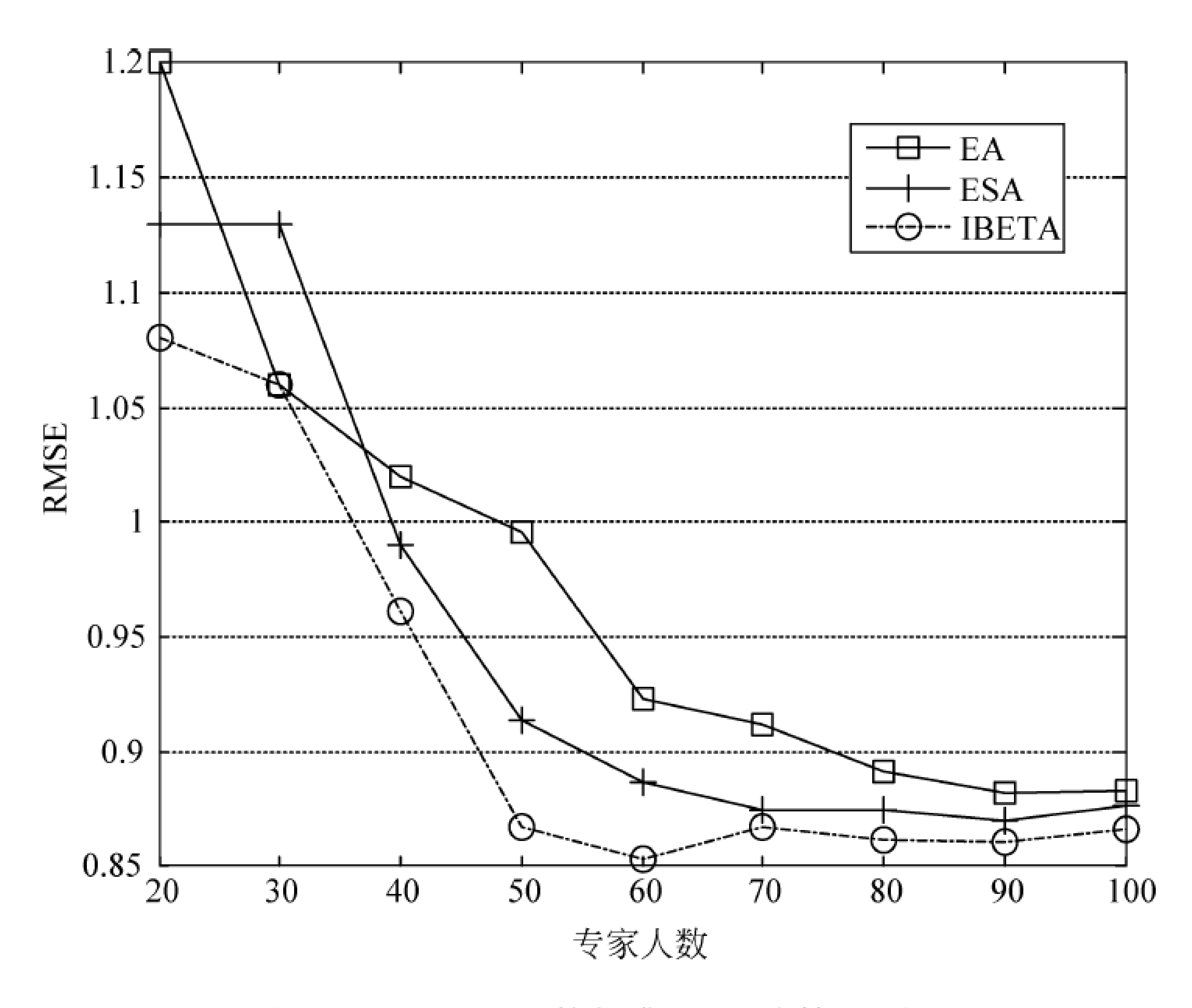


图 12-9　Epinion 数据集上三种算法对比

本章小结

本章研究了专家算法的产生与改进，IBETA 算法在 ESA 算法和 EA 算法的基础上加入专家信任度用户、项目偏置信息。实验表明，改进后的带偏置专家信任协同过滤推荐算法在稀疏用户的预测准确性方面有较大提高，但是，在算法的改进过程中专家信任指标的融合还不够完善。所以，下一步工作的重心将放在信任的动态调整及建立有效的信任传递机制，使信任度量更加合理。

参考文献

[1] Adomavicius G, Tuzhilin A. Toward the Next Generation of Recommender System[J]. IEEE Transactions on Knowledge & Data Engineering, 2005, 17(6): 734-749.

[2] Zhu Yang-yong, Sun Qian. Research progress of recommendation system[J]. Computer science and exploration, 2015, 9(5): 513-525.

[3] Mcafee A, Brynjolfsson E. Big data: the management revolution[J]. Harvard Business Review, 2012, 90(10): 60-66.

[4] Piao C H, Zhao J, Zheng L J. Research on entropy-based collaborative filtering algorithm and personalized recommendation in e-commerce[J]. Service Oriented Computing & Applications, 2009, 3(2): 147-157.

[5] Yu Feng-quan, Wang Xu-ming, Xie Yan-hong. Comparison of data smoothing algorithms for flight data processing[J]. Command control and simulation, 2015(1): 116-119.

[6] Deng Ai-lin, Zhu Yang-yong, Shi Bo-le. Collaborative filtering recommendation algorithm based on project score predicts[J]. Journal of Software, 2003, 14(9): 1621-1628.

[7] Wang Q M, Liu X, Zhu R, et al. A New Personalized Recommendation Algorithm of Combining Content-based and Collaborative Filters[J]. Computer &Modernization, 2013, 1(8): 64-67.

[8] Hwang W S, Lee H J, Kim S W, et al. On using category experts for improving the performance and accuracy in recommender systems[C]. ACM International Conference on Information and Knowledge Management, 2012: 2355-2358.

[9] Liu Qiang. Research on the key algorithm in collaborative filtering recommendation system [D]. Hongzhou: Zhejiang University, 2013.

[10] Cho J, Kwon K, Park Y. Collaborative Filtering Using Dual Information Sources[J]. IEEE Intelligent Systems, 2007, 22(3): 30-38.

[11] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering [C]//Fourteenth Conference on Uncertainty in Artificial Intelligence, 2013: 43-52.

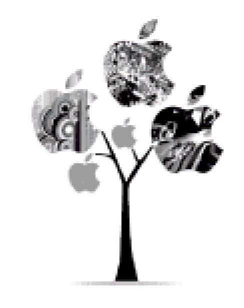
[12] Gouya G, Arrich J, Wolzt M, et al. Antiplatelet treatment for prevention of cerebrovascular events in patients with vascular diseases[J]. A journal of Cerebral Circulation, 2014, 45(2): 492-503.

[13] Marinho L B, Hotho A, Jäschke R, et al. Baseline Techniques[M]. US: Springer US, 2012.

[14] Peng Fei, Deng Jiang-hao, Liu Lei. Add user ratings offset recommendation system model

[J]. Journal of Xi'an Jiaotong University,2012,46(6): 74-78.
[15] Shen Li-men,Wang Li-hua,Li Feng. An adaptive trust model based on time series analysis in opportunistic networks [J]. Journal of Chinese Computer Systems, 2015, 36 (7): 1553-1558.
[16] Hernando A,Lazaro J,Gil E,et al. Inclusion of respiratory frequency information in heart rate variability analysis for stress assessment. [J]. IEEE Journal of Biomedical & Health Informatics,2016,20(4): 1016-1025.
[17] Shambour Q, Lu J. An effective recommender system by unifying user and item trust information for B2B applications[J]. Journal of Computer & System Sciences, 2015, 81 (7): 1110-1126.
[18] Zheng X L,Chen C C,Hung J L,et al. A Hybrid Trust-Based Recommender System for Online Communities of Practice[J]. IEEE Transactions on Learning Technologies,2015,8 (4): 345-356.
[19] Golbeck J. Personalizing applications through integration of inferred trust values in semantic web-based social networks [J]. Proceedings of Semantic Network Analysis Workshop,2005.
[20] Wu Z,Yu X,Sun J. An Improved Trust Metric for Trust-Aware Recommender Systems [C]//International Workshop on Education Technology and Computer Science. IEEE, 2009: 947-951.
[21] 秦继伟,郑庆华,郑德立,等. 结合评分和信任的协同推荐算法[J]. 西安交通大学学报, 2013,47(4): 100-104.
[22] 王海艳,张大印. 一种可信的基于协同过滤的服务选择模型[J]. 电子与信息学报,2013,35 (2): 349-354.
[23] Zeng J, Gao M, Wen J, et al. A Hybrid Trust Degree Model in Social Network for Recommender System [C]//Iiai, International Conference on Advanced Applied Informatics. IEEE,2014: 37-41.
[24] 朱强,孙玉强. 一种基于信任度的协同过滤推荐方法[J]. 清华大学学报(自然科学版), 2014(3): 360-365.
[25] Ma T, Zhou J, Tang M, et al. Social Network and Tag Sources Based Augmenting Collaborative Recommender System[J]. Ieice Transactions on Information & Systems, 2015,E98. D(4): 902-910.
[26] [48]Shen X, Long H, Ma C. Incorporating trust relationships in collaborative filtering recommender system[C]//IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, NETWORKING and Parallel/distributed Computing. IEEE, 2015: 1-8.
[27] Parnes P,Synnes K,Schefström D. mTunnel: A multicast tunneling system with a user-based quality-of-service model[M]. Springer,2016,1309: 87-96.
[28] Li D, Chen C, Lv Q, et al. An algorithm for efficient privacy-preserving item-based collaborative filtering[J]. Future Generation Computer Systems,2016,55: 311-320.
[29] 潘骏驰,张兴明,汪欣. 融合用户可信度的改进奇异值分解推荐算法[J]. 小型微型计算机系统,2016,37(10): 2171-2176.
[30] Chen C, Zheng X, Zhu M, et al. Recommender System with Composite Social Trust Networks[J]. International Journal of Web Services Research,2016,13(2): 56-73.

第 13 章 一种改进专家信任的协同过滤推荐算法



本章针对传统基于用户的协同过滤推荐算法较少考虑信任对象所处环境的实时变化，提出一种结合社交网络的专家信任推荐算法。为更好地量化对象之间的信任度，首先利用专家的评价可信度、活跃度、评价偏差度等量化因子计算得到专家的信任值。其次在评分形成的过程中与近邻算法相融合，明确用户与“专家”和“近邻”的偏好，当可选专家人数小于预先设定的阈值时，利用协调因子动态调整近邻算法与改进专家算法的权重，以便获得更加客观的项目评分。最终实验结果表明，在不同大小的 MovieLens 数据集上相比于传统的推荐算法，本章提出的算法在实时推荐预测准确度方面有显著提高。

13.1 引言

随着互联网信息的急剧增长，“信息过载”的现象越来越严重，推荐系统作为解决信息过载的有效方案，目前无论工业界还是学术界，对如何完善协同过滤推荐算法都展开了深入的研究。近邻模型作为协同过滤推荐系统的关键算法，因其原理简洁且具有较低的时间复杂度，逐渐成为当今推荐系统中被广泛使用的算法之一。

专家模型作为近邻模型的可选方案之一，自 2012 年被 Amatrin[4]提出之后受到广泛关注。2013 年，Kagita 提出了“明星用户”算法，该算法在推荐过程中仅使用“明星用户”的偏好，并没有加入项目的相似度，当可选明星用户数达不到预先设定的阈值时，单一考虑“明星用户”评分会导致该算法的推荐准确度不高。2014 年 Cho 在利用专家算法形成推荐的过程中，同时兼顾专家与近邻的项目评分，提高了专家算法推荐的准确性及合理性。2015 年 Won-Seok Hwang 为了使推荐准确性进一步提高，提出了专家算法与协同过滤技术的结合。Cho 和 Won-Seok Hwang 提出的算法，虽然在一定程度上缓解了可选专家人数达不到预先设定阈值时的推荐问题，但在形成推荐的过程中，把专家对项目的评分同等看待显然有失偏颇，因为专家的专业水平有高有低，可信度有大有小。

针对传统专家算法在评分形成的过程中同等看待专家评分这一现象，本章提

出了一种结合社交网络的专家信任推荐算法——IETA 算法(Improve Expert Trust Algorithm),该算法利用相似度计算公式得到用户间的相似度矩阵,同时利用项目类别矩阵确定待评分项目类别,进而选出该类别的专家,通过专家信任指标计算专家信任度。通过赋予与用户相似的专家不同的信任值,利用专家的项目评分及信任值形成最终的推荐。

13.2　标注与相关工作

13.2.1　标注

为本章下文表述方便,在此对文中使用的标注作统一说明,如表 13-1 所列为用户—项目评分矩阵,用 m 个用户和 n 个项目的评分矩阵 R 来标示所有用户对所有项目的评分,范围是[1,5]。

表 13-1　用户—项目评分矩阵

u	i					
	i_1	i_2	i_3	…	i_{n-1}	i_n
u_1	5	0	3	0	5	0
u_2	0	0	0	0	0	4
u_3	2	0	0	3	0	0
⋮	0	0	5	1	0	0
u_{m-1}	2	0	0	4	0	0
u_m	0	4	0	0	0	1

如果用户 u 未对项目 i 评分,那么值为 0,如式(13-1)所示。表 13-2 所列为项目—类别矩阵。

$$R_{u,i}=\begin{cases}R_{u,i}, & \text{评分}\\ 0, & \text{未评分}\end{cases} \tag{13-1}$$

式中:$R=\{r_{u,i}\}(1\leqslant u\leqslant m,1\leqslant i\leqslant n)$,其中 $R_{u,i}$表示用户 u 对项目 i 的评分。

表 13-2　项目—类别矩阵

i	c					
	c_1	c_2	c_3	…	c_{18}	c_{19}
i_1	1	0	1	0	1	0
i_2	0	0	0	0	0	1
i_3	1	0	0	1	0	0
⋮	0	0	1	1	0	0
i_{n-1}	1	0	0	1	0	0
i_n	0	1	0	0	0	1

如果项目 i 属于类别 c，那么值为 1，如式(13-2)所示。

$$T_{i,c}=\begin{cases}1, & \text{属于}\\0, & \text{不属于}\end{cases} \tag{13-2}$$

式中：$T=\{t_{i,c}\}(1\leqslant i\leqslant n,1\leqslant c\leqslant 19)$，其中 $T_{i,c}$ 表示项目 i 是否属于类别 c。

13.2.2 近邻模型

近邻模型的基本原理是寻找 k 个近邻近似替代当前用户。在协同过滤推荐算法中，近邻模型分为基于用户的近邻模型和基于项目的近邻模型。用户近邻模型分为寻找近邻和形成推荐两部分，为了解决寻找近邻的问题，首先需要找到一种方式来表示用户之间的距离关系，通常用户之间的邻近程度有以下两种计算方式。

1. 余弦相似度

当用户共同评分项目数量较少时，该公式计算用户之间相似度偏差会比较大。当用户评分项目数量达到一定值时，即使用户对每个项目的评分值差别较大，该算法也能输出较高的相似值，余弦相似度公式如式(13-3)所示。

$$s(u,v)=\frac{\sum_{i\in I'}r_{ui}r_{vi}}{\sqrt{\sum_{i\in I'}r_{ui}^2\sum_{i\in I'}r_{vi}^2}} \tag{13-3}$$

式中：r_{ui} 与 r_{vi}——用户 u 与用户 v 对项目 i 的评分；

I'——用户 u 与 v 的共同评分集合。

2. 改进余弦相似度

改进余弦相似度同样没有解决 I' 较少时，相似度计算偏差较大的问题。但是 I' 的值在可接受范围内时，改进余弦相似度体现出了较好的性能。改进余弦相似度公式如式(13-4)所示。

$$s(u,v)=\frac{\sum_{i\in I'}(r_{ui}-\bar{r}_u)(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I'}(r_{ui}-\bar{r}_u)^2}\sqrt{\sum_{i\in I'}(r_{vi}-\bar{r}_v)^2}} \tag{13-4}$$

式中：$\bar{r}_u$ 与 $\bar{r}_v$——用户 u 与用户 v 评分的均值；

I'——用户 u 与用户 v 共同评分项目集合。

13.2.3 专家算法

定义 13.1 对于 A 类项目，专家 E_c 定义如式(13-5)所示。

$$|I_u|\leqslant|I_v| \quad (\forall u\in U-E_c,\forall v\in E_c) \tag{13-5}$$

式中：I_u——用户 u 评价过的所有项目集合；

I_v——专家 v 评价的项目集合；

U——所有用户集合。

统计每个用户评价 A 类项目数量，当用户评价项目的数量使式(13-5)成立

时，该用户被定义为“专家”。

专家算法在形成之初由寻找专家与生成推荐值两部分构成，但是在实际操作中这种结构推荐效果不理想，随着越来越多的专业人员参与研究和改进，专家算法目前由寻找专家、计算专家与用户的相似值以及生成推荐值三部分组成。

1. 寻找专家

根据待评分项目和项目—类别矩阵，确定该项目所属类别，对所有的用户计算评价该类别所有项目的次数，从大到小依次排列，由专家的定义及预先设定阈值确定专家人数。

2. 生成推荐值

Pham 的“专家算法”中，在计算预测评分时，只考虑与当前用户相似度比较高的专家建议，采用无条件相信专家的策略——EA(Expert Algorithm)。在度量专家与用户的相似度时采用式(13-4)，最终评分预测如式(13-6)所示。

$$p_{u,i,c} = \bar{r}_{u,c} + \frac{1}{k}\sum_{v\in E_c}(r_{v,i} - \bar{r}_{v,c}) \tag{13-6}$$

式中：$p_{u,i,c}$——当前用户对属于 c 类电影 i 的预测评分；

$\bar{r}_{u,c}$——用户 u 对 c 类电影评分的平均值；

$r_{v,i}$——专家 v 对项目 i 的评分；

$\bar{r}_{v,c}$——专家 v 对 c 类项目评分的平均值；

E_c——c 类项目专家集合。

需要预测的项目属于多个类别时，计算评分值如式(13-7)所示。

$$p_{u,i} = \frac{1}{|c_i|}\sum_{c\in c_i} p_{u,i,c} \tag{13-7}$$

式中：c_i——项目所属的类数；

$p_{u,i,c}$——每一类预测的值。

以上预测评分算法的运行时间比较短，但是它在预测准确性方面表现一般。此算法在项目确定的情况下，对于不同用户的预测分数几乎是一样的，因为专家的选择没有考虑当前用户，只考虑了当前用户需要预测的项目。

13.3　改进专家算法

从专家算法提出至今，许多研究人员都围绕着利用专家与用户、项目的关系提升推荐准确度。但是现实生活中人们在参考权威人士的意见时，必然要考虑权威人士的可信度、专业度、偏差度等影响因子。到目前为止推荐系统并没有对“信任”给出一个具体的概念。在可查资料中，信任是指接受推荐者对提供推荐者特定行为的主观可能性预测。在社交网络中信任需要考虑的因素更多，完整地考虑各个方面难度很大且通常没有必要，在面对同一个用户时，只需要对该用户所处的情形

进行相应的加强和减弱，以便于对对象之间的信任程度进行较好的量化。

13.3.1 重要概念

在推荐系统中存在着各种各样的数据，其中包括评分数据、项目属性数据、用户属性数据等，这些数据基本构成了本章需要的信任度量情境。充分考虑专家及用户所在环境，本章用以下定义量化专家信任中涉及的重要概念。

定义 13.2 专家评价可信度

一个专家评价的项目数量越多，可以从一定程度上反映出其评价项目的质量、可信度，度量专家评价可信度如式(13-8)所示。

$$D_u = \frac{Q_u}{\max(Q_{\text{all}})} \tag{13-8}$$

式中：Q_{all}——所有用户；

Q_u——专家 u 评价过的所有项目的集合；

$\max(Q_{\text{all}})$——所有专家中评价项目最多的数量。

定义 13.3 专家专业度

咨询专家意见之前，人们通常会考虑专家的专业度，专家并不是对所有种类的项目都具有全面的专业知识，在某种情况下，一名专家显然只会对一个或者很少种类的项目上投入比较多的精力，具体表现为在某一类项目上评价比较多的项目，因此专家专业度如式(13-9)所示。

$$R_u = \frac{T_{ui}}{T} \tag{13-9}$$

式中：T_{ui}——专家已评价且属于某一种类的所有项目集合；

T——系统中获得过用户评价且属于这一主题的所有项目集合。

定义 13.4 专家评价偏差度

专家计算的预测评分与真实评分之间的差值为专家评价偏差度，如式(13-10)所示。

$$P_u = \frac{Z_u}{Q_u} \tag{13-10}$$

式中：Z_u——最小偏差项目的集合。

在计算 Z_u 时，利用项目评分的平均值 δ 表示项目的真实质量，专家的评价的偏差如果小于 δ，则把此评分项目加入到 Z_u 中。对专家 u 及项目 i，如果 $|r_{u,i}-\overline{r_i}|\leqslant\delta$ 成立，则 $i\in\delta$，通过实验验证 δ 如果太小则 Z_u 趋于零，计算就没有太大意义；太大会使 δ 趋于 1，计算效果不理想。本实验 δ 取为 0.37。

基于以上表述专家的 D_u，P_u，R_u，权重系数 w_1，w_2，w_3，计算专家信任值如式(13-11)所示。

$$TR_u = w_1 \cdot D_u + w_2 \cdot P_u + w_3 \cdot R_u \tag{13-11}$$

为更直观地体现专家信任对评分预测的影响，利用原始专家算法与信任相结

合进行调整参数值。本章所涉及的三种信任指标相互独立，权重系数的最优值选取采用固定两个调整另外一个的策略（如设置一个适当的 RMSE 值，初始化 w_1、w_2、w_3，把专家信任值与式(13-6)、式(13-7)结合产生预测，计算此时 RMSE 值，如果当前 RMSE 值大于设定 RMSE 值，固定 w_2、w_3 更新一次 w_1，记录 w_1 达到最优 RMSE 时的值。同理可得最优 w_2、w_3），最后归一化处理三个参数得到最终权重系数值。

13.3.2　评分形成

2015 年 Ho-Jong Lee 提出了专家与相似度结合的算法——ESA（Expert Similarity Algorithm），在预测项目评分时 ESA 算法运用经典的评分预测式(13-6)与式(13-12)相结合，进一步提升了算法的推荐准确度。

$$p_{u,i}=\frac{\sum_{c\in c_i}p_{u,i,c}\cdot f_{u,c}}{\sum_{c\in c_i}f_{u,c}} \tag{13-12}$$

式中：$f_{u,c}$——该项目的专家专业度。

式(13-12)与式(13-7)相比其进步在于专家对每一类项目的评分在最终预测评分时权重不同。本章在预测评分时改进了 ESA 算法采用公式(13-13)来求解。

$$p_{u,i}=\frac{1}{\sum_{c\in C_i}TR_u}\times\sum_{c\in C_i}\left(\bar{r}_{u,c}+\frac{\sum_{v\in E_c}(r_{v,i}-\bar{r}_{v,c})\cdot s(u,v)}{\sum_{v\in E_c}s(u,v)}\right)\cdot TR_u \tag{13-13}$$

式中：TR_u——专家可信度；

C_i——当前项目所属类别总数。

此类算法对于每个电影类别的专家评分，根据专家在此类别评价项目中信任值，加权计算预测评分，有效避免了出现不同类别专家对项目的评分同等对待的现象，充分考虑用户与专家之间的相似度及专家信任值，根据专家信任值赋予不同专家不同的权重，在一定程度降低了预测误差。

当专家人数达不到设定的阈值时，近邻算法作为补充，两者结合发挥两种算法的优点，两者结合可用式(13-14)表示。

$$p_{ui}=\overline{r_u}+\left(\alpha\times\frac{\sum_{v\in s_u}(r_{vi}-\bar{r}_v)\times s(u,v)}{\sum_{v\in s_u}|s(u,v)|}+(1-\alpha)\times\frac{\sum_{z\in s_u'}(r_{zi}-\bar{r}_z)\times TR_z}{\sum_{z\in s_u'}(TR_z)}\right) \tag{13-14}$$

式中：r_{vi}——用户 v 对项目 i 的评分；

$\bar{r}_u$——当前用户 u 对所有项目的平均评分；

$s(u,v)$——用户 u 与用户 v 的相似度；

s_u——近邻集合；

s'_u——专家集合。

当专家人数达不到预先设定的阈值时，专家对项目的评分预测具有一定的偶然性，此时的专家集合不能全面代表整体对项目的评分。本章将此时的专家视为是某种意义上的近邻，采用 α 作为协调因子，协调专家与近邻预测评分，当专家人数为零时，此时式(13-14)退化为近邻算法评分预测公式，如式(13-15)所示。

$$\alpha = \frac{s(u,v)^2}{s(u,v)^2 + TR_u^2} \tag{13-15}$$

13.3.3 算法描述

算法 13-1 基于社交网络中改进专家信任的协同过滤推荐算法(IETA)

输入：评分矩阵 R 及项目类别矩阵 T，RMSE 阈值 0.98。

输出：预测矩阵 R_{pred}。

步骤 1：数据预处理——随机产生 w_1,w_2,w_3。

步骤 2：利用式(13-3)计算训练集中用户之间的相似度，形成相似度矩阵。

步骤 3：根据欲评分项目 i 和项目类别矩阵，确定项目 i 所属类别的专家。

步骤 4：根据评分偏差修正一次 w_1,w_2,w_3，直至出现最优 RMSE 值。

```
for i=1：Round
  根据式(13-12)计算预测值并计算 RMSE 值。
    if(当前 RMSE 值>设定 RMSE 值)
      修正一次 w1,w2,w3 的值；
    else
      记录当前 RMSE 值及对应的可调参数；
    end
  i++；
end
  return(对应最优 RMSE 值的可调参数并归一化)；
```

步骤 5：根据步骤 3 寻找到的专家及式(13-11)，当可选专家人数不少于设定专家阈值 30 时，利用式(13-13)转到步骤 7，否则转到步骤 6。

步骤 6：根据 α 的值及式(13-15)转到步骤 7。

步骤 7：产生预测矩阵 R_{pred}。

算法结束

13.4 实验结果与分析

13.4.1 数据集

实验数据集是由美国 Minnesota 大学 GroupLens Research 实验室提供 MovieLens 数据集，该数据集包含匿名用户对电影的评分，其中每个用户至少评价

了其中 20 部电影，评分值的范围是 1～5，1 表示最低评分，5 表示最高评分，0 表示用户没有评价该电影。除了评分数据外，该数据集中包含用户、项目的属性，例如用户的性别、年龄、职业以及项目的名称、上映年份、风格流派等，其中用户电影的风格流派、评分数据是本章实验所需要的。

13.4.2　评估标准

评估推荐系统预测准确性的标准分为决策精度标准和统计精度标准两类。本章采取了对特大或特小误差反应敏感的均方根误差（RMSE）。在推荐系统中 RMSE 作为一种常用度量误差标准被广泛使用，其原理是通过计算用户关于项目的预测值与真实值之间的偏差平方和与用户个数 n 比值的平方根，如式（13-16）所示。

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i}^{n}(x_i-x_o)^2}{n}} \tag{13-16}$$

式中：x_i——预测值；

x_o——与预测值对应的真实值。

13.4.3　实验结果与分析

1. 相似度实验对比

为了得到较佳的实验结果，本章分别用 13.2 节中余弦相似度和改进的余弦相似度计算方法计算项目之间的相似度，如图 13-1 所示为两种相似度分布对比图。从图 13-1 可以看出，余弦相似度分布较为均匀，改进的余弦相似度分布更具个性化。在［0，1］范围内，运用余弦相似度得到的相似度值分布在［0.0，0.6］，平均达到 89.10％，分布过于分散。而改进的余弦相似度得到的相似度值主要分布在［0.0，0.4］，平均达到 80.15％，因为通过项目的评分值减去用户评分的平均值，均衡了用户的评分尺度不一问题，更真实地反映出项目的差异特征，即用户的个性化选择。所以，根据改进的余弦相似度计算方法可以得到较高质量的推荐。基于以上分析，本章采用改进的余弦相似度计算方法进行度量。

2. 用户可信度指标分布与分析

用户评价可信度、用户专业度及专家评价偏差度的分布情况如图 13-2 所示。在 MovieLens 100k 数据集中，用户可信度主要分布在［0，0.4］，其中 56.4％的用户评价可信度分布在［0，0.1］，剩余的用户评价可信度在其他区间都有分布，说明了少数用户评价可信度能在全体的用户评价可信度中体现个性化的特质；同时，用户专业度主要分布在［0，0.3］，其中 58.21％的专家分布在［0，0.2］，再次说明了只有少数用户对某些类别的项目比较专业。从图 13-2 可以看出，用户的评价偏差度分布几乎成正态分布，评价偏差度分布在［0.2，0.6］的用户所占比例为 77.6％，说

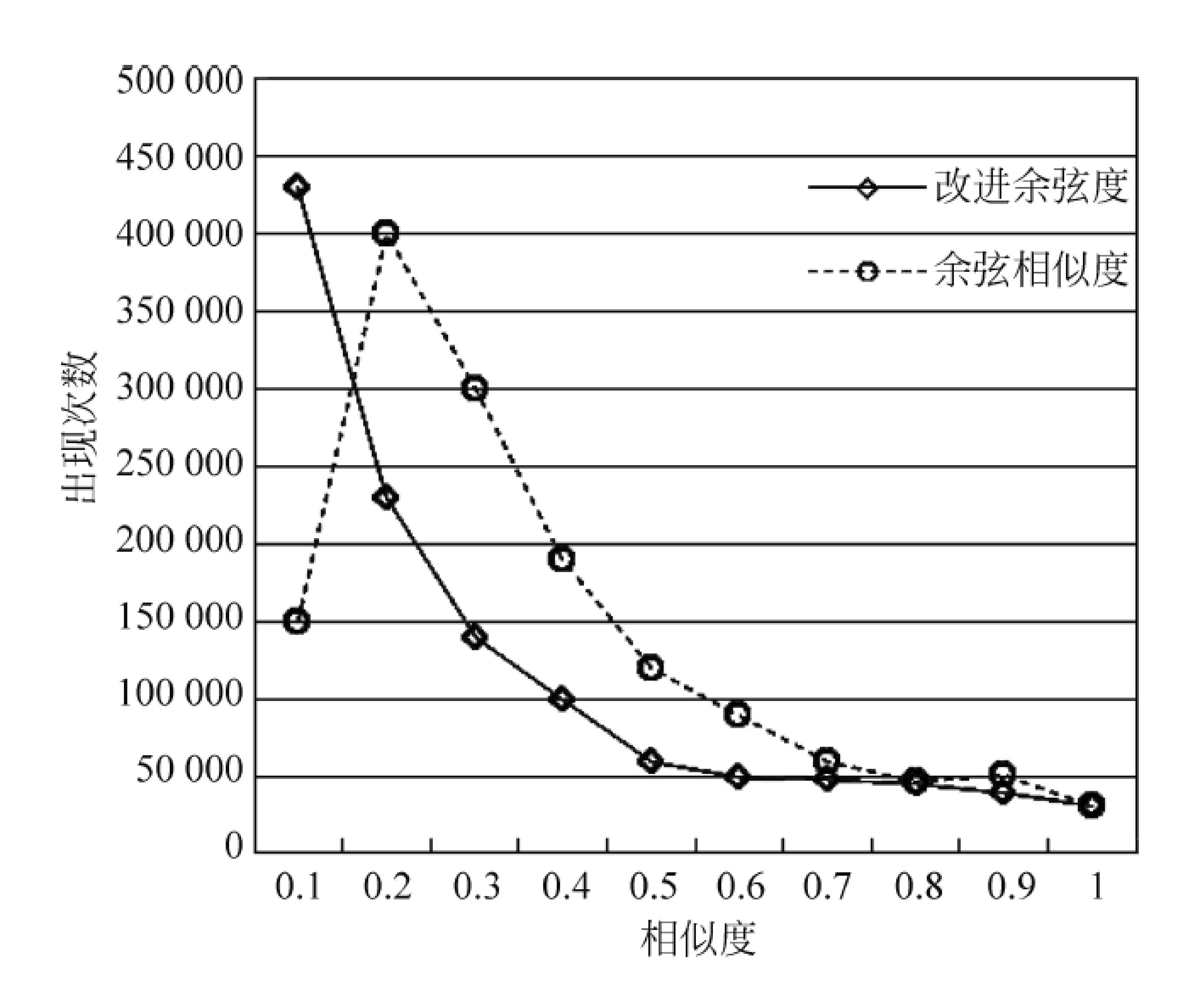


图 13-1　两种相似度分布对比图

明大多数用户的评价偏差度比较高(评价比较接近真实评分),以上足以说明选定专业用户以后(专家),该专家对项目的评分信任度可以由评价可信度、专业度、评价偏差度三种指标体现。

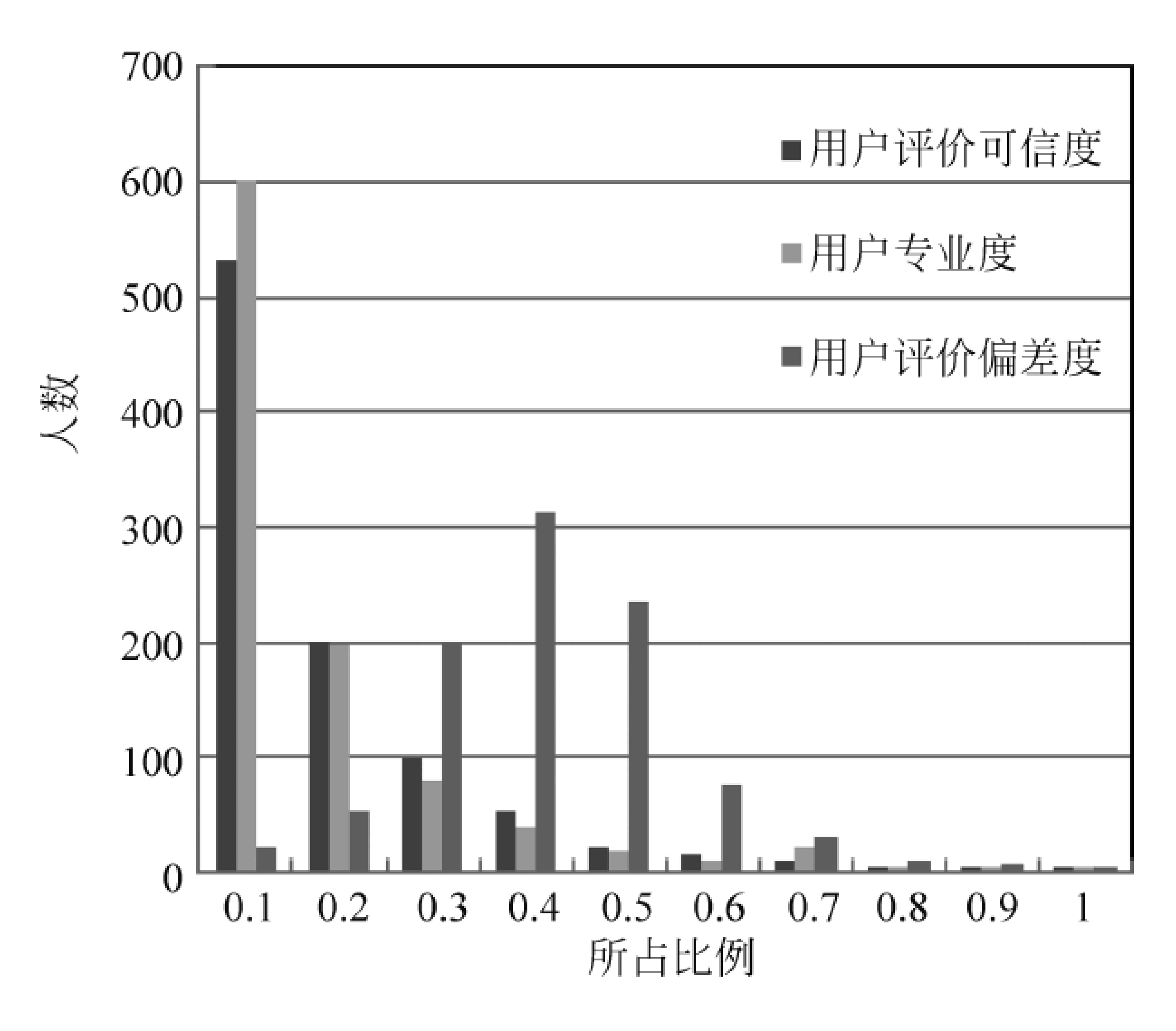


图 13-2　评价指标分布图

3. 专家可信度分布与分析

MovieLens 100k 数据集由 19 种流派的电影组成,图 13-3 为专家可信度分布图。选择其中一种流派的电影并计算该流派专家的信任值,在实验开始阶段,w_1,w_2,w_3 是初始化值,所以专家可信度值比较低。随着调整次数的增加,专家信任值从 100 000 次调整的 0.12 提升到 0.63 共经过了 400 000 次的调整,从 400 000 次

开始专家信任值稳定在[0.63,0.69],此时归一化后得出 w_1,w_2,w_3 值分别为0.31,0.46,0.23,在后续计算专家对此类别项目评分时,可以直接利用此训练后的系数值;对于不同类别的专家信任因子系数值不同,可使用同样方法训练得到。

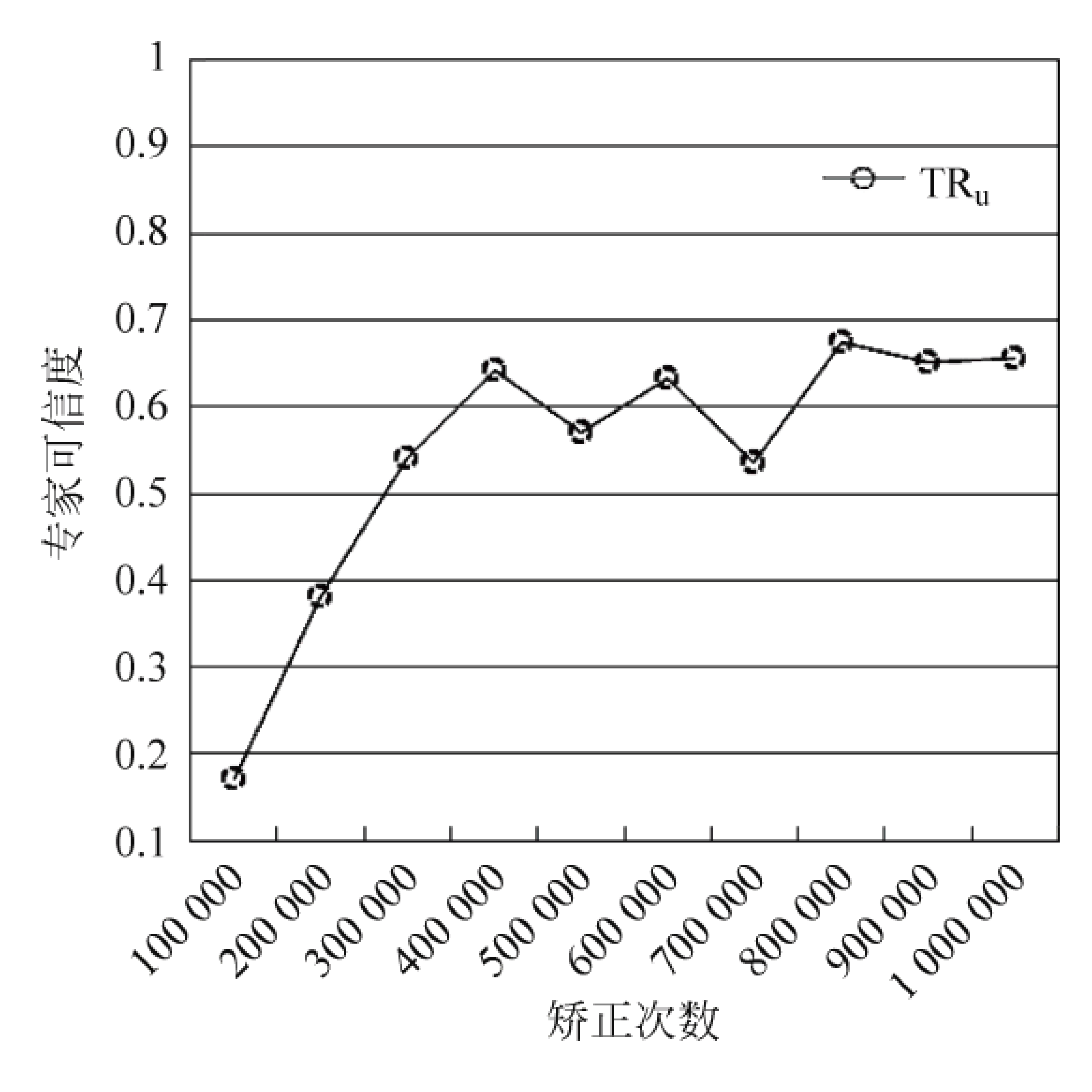


图 13-3　专家可信度分布图

4. 专家近邻融合

专家人数达不到预先设定的阈值时,专家近邻算法融合对比如图 13-4 所示。未融合专家与近邻算法的下降速度较慢,原因是在开始阶段每个类别的专家可选人数在达不到预先设定的阈值时,专家评价项目的准确性有待进一步提高,在计算此类项目的评分时把近邻算法与专家算法结合,一方面发挥了近邻算法的快速收敛性能,另一方面把没有达到符合人数要求的专家与近邻融合补充了近邻算法的不确定性(近邻可能从来没有评价过此类项目),因此本章在专家人数达不到预先设定的阈值时选择了近邻算法与专家算法的融合。

5. 专家信任算法对比

本节选取对比 EA、ESA 与本章提出的 IETA 算法在专家数量不同情况下的推荐精确度。实验以 RMSE 为评估标准,并分别基于 Movielens_100k 数据集和 Movielens_1M 数据集进行。

图 13-5 是基于 MovieLens 100k 数据集所绘曲线。可以明显看出,在实验开始阶段随着专家人数的增加,IETA 算法的 RMSE 下降速度最快;随着专家人数的增加,ESA 算法的 RMSE 值在 0.88 左右浮动,与 EA 算法相比,随着专家专业度与用户相似度的加入,不再直接使用专家推荐值,而是根据专家专业度与专家与用户的相似度计算推荐结果,进一步提升了推荐结果的合理性。

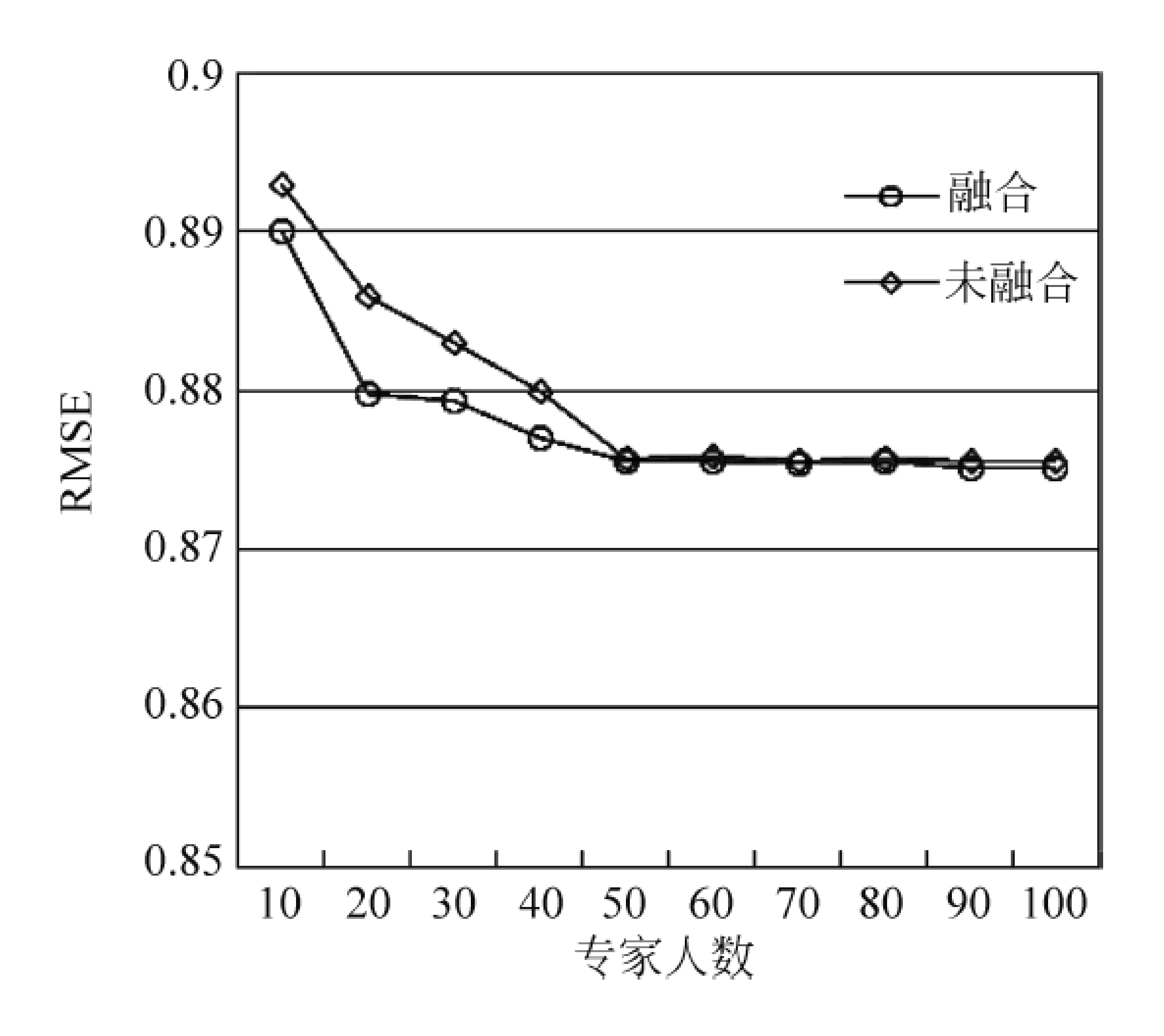


图 13-4　专家近邻算法融合对比图

IETA 算法把专家信任与用户相似度加入到 ESA 算法中，RMSE 值在专家人数达到 50 人时达到最小值 0.875，在专家人数进一步增加时该算法的 RMSE 值趋于稳定。以上实验对比表明，IETA 算法在推荐准确性及合理性上均有提高。

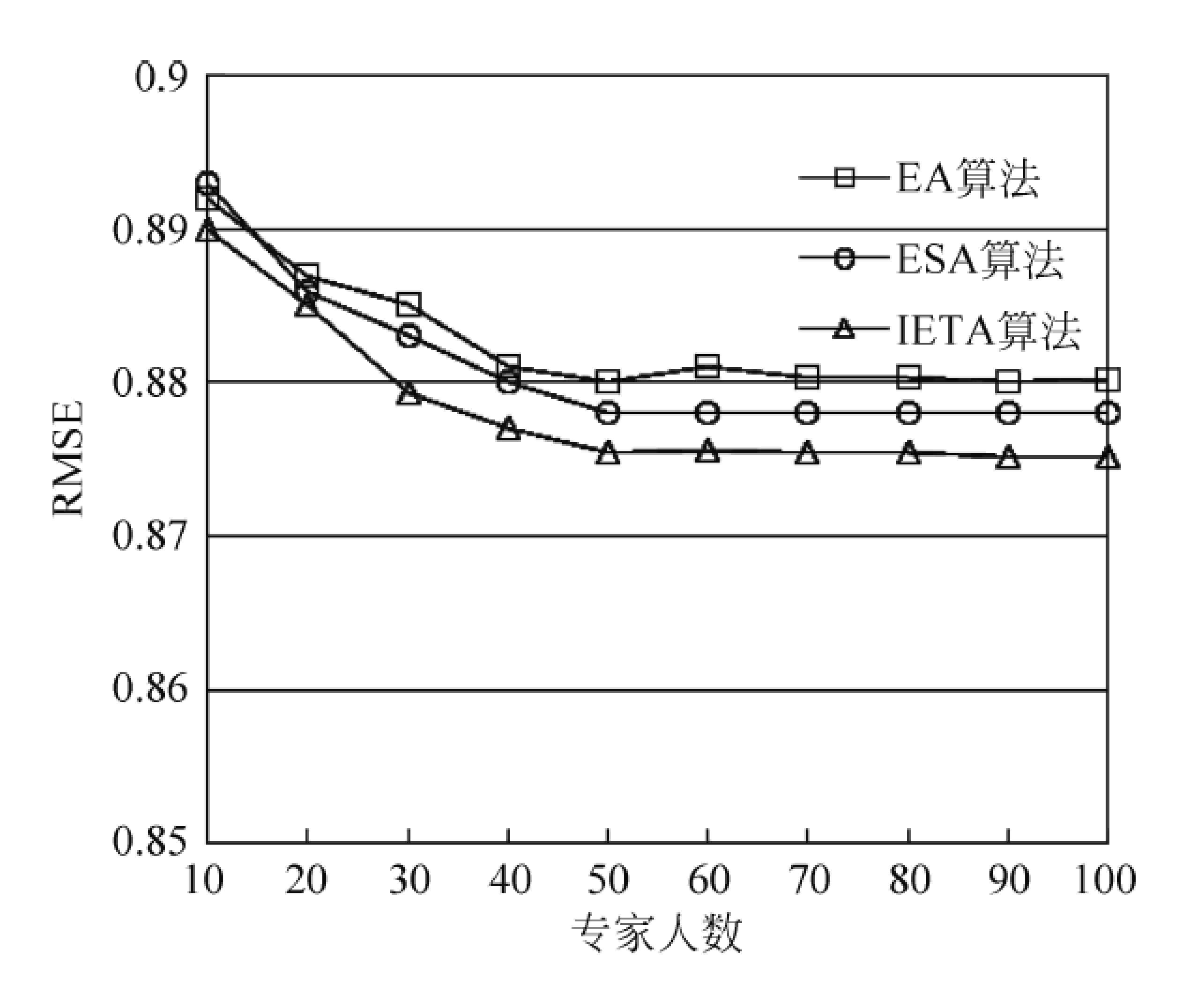


图 13-5　基于 MovieLens 100k 数据集

图 13-6 为基于 MovieLens 1M 数据集，表明在实验数据集为 1M 时，IETA 算法在训练集内的训练会更加充分，EA 与 ESA 算法的计算结果没有随着数据量的增加而有所改变，说明其扩展性表现不佳。但本章提出的 IETA 算法的 RMSE 提升了 5%，对比结果进一步说明了本章提出的算法具有更好的适应性和扩展性。

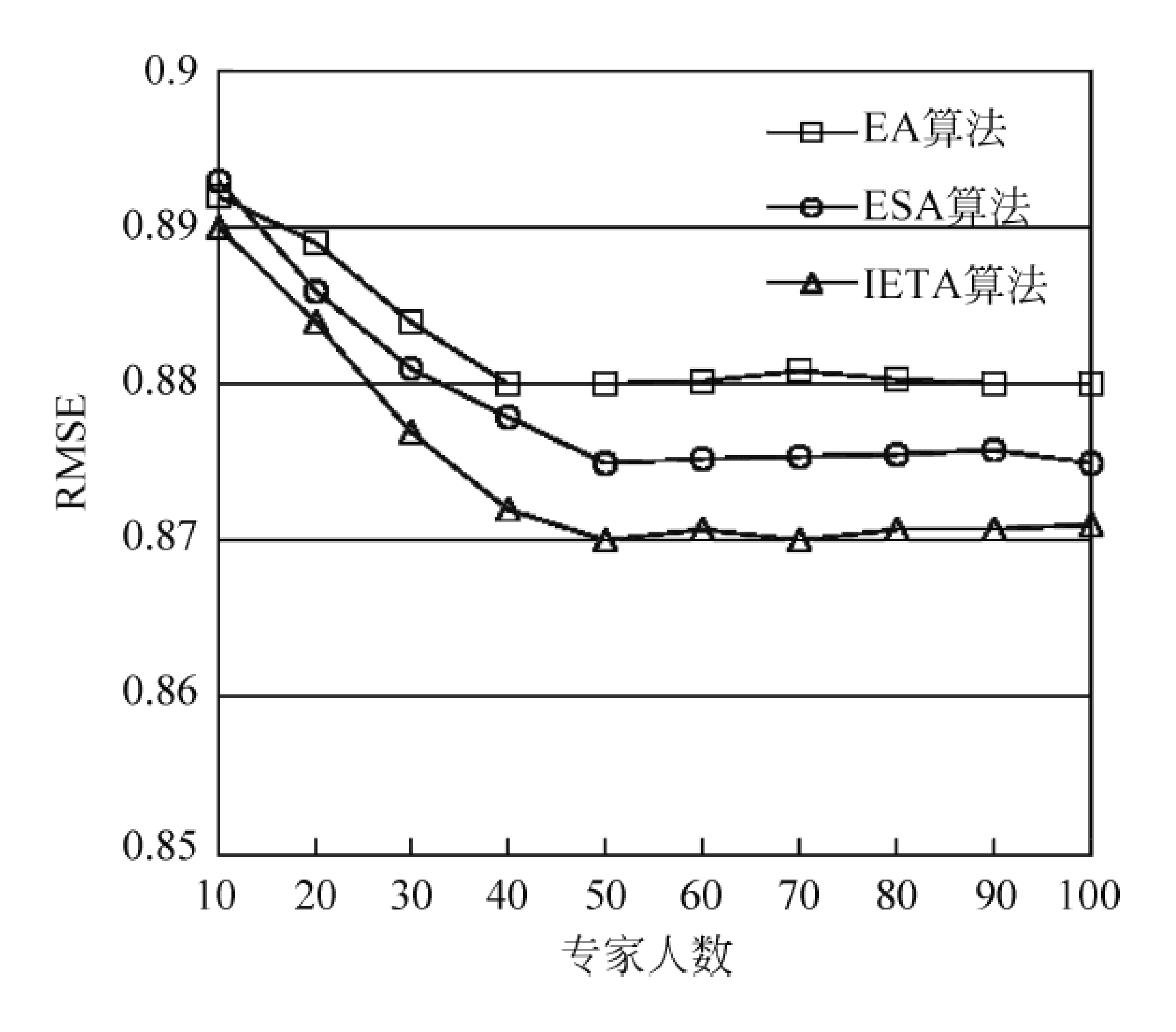


图 13-6 基于 MovieLens 1M 数据集

本章小结

本章研究了专家算法的产生与改进，IETA 算法在 ESA 算法和 EA 算法的基础上加入专家信任度。实验表明，改进后的专家信任协同过滤推荐算法不仅有效提高了推荐系统的推荐精度，而且随着实验数据集的增大展现出了良好的扩展性。但是，在算法的改进过程中专家信任指标的融合还不够完善。所以，下一步工作的重心将放在信任的动态调整及建立有效的信任传递机制，使信任度量更加合理。

参考文献

[1] Piao C H, Zhao J, Zheng L J. Research on entropy-based collaborative filtering algorithm and personalized recommendation in e-commerce. SOCA[J]. Service Oriented Computing & Applications, 2009, 3(2): 147-157.

[2] Adomavicius G, Tuzhilin A. Tuzhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions[J]. IEEE Transactions on Knowledge & Data Engineering, 2005, 17(6): 734-749.

[3] Lu Z, Shen H. A Fast Algorithm to Build New Users Similarity List in Neighbourhood-Based Collaborative Filtering[M]. Advances in Parallel and Distributed Computing and Ubiquitous Services. Springer Singapore, 2015.

[4] Amatriain X, Lathia N, Pujol J M, et al. The Wisdom of the Few A Collaborative Filtering Approach Based on Expert Opinions from the Web[C]//International Acm Sigir Conference on Research & Development in Information Retrieval, 2009: 532-539.

[5] Kagita V R, Padmanabhan V, Pujari A K. Precedence Mining in Group Recommender Systems[M]. Pattern Recognition and Machine Intelligence. Springer Berlin Heidelberg,

2013：701-707.
[6] Kardan A,Aziz M,Shahpasand M. Adaptive systems：a content analysis on technical side for e-learning environments[J]. Artificial Intelligence Review,2015,44(3)：365-391.
[7] Hwang W S,Lee H J,Kim S W,et al. Efficient recommendation methods using category experts for a large dataset[J]. Information Fusion,2015,28(C)：75-82.
[8] Zamanzad G F,Claesen J,Burzykowski T,et al. Comparison of the Mahalanobis distance and Pearson's χ^2 statistic as measures of similarity of isotope patterns. [J]. Journal of the American Society for Mass Spectrometry,2014,25(2)：293-296.
[9] Cho J,Kwon K,Park Y. Collaborative Filtering Using Dual Information Sources[J]. IEEE Intelligent Systems,2007,22(3)：30-38.
[10] Hwang W S,Lee H J,Kim S W,et al. Efficient recommendation methods using category experts for a large dataset[J]. Information Fusion,2015,28(C)：75-82.
[11] Gohari F S,Haghighi H,Aliee F S. A semantic-enhanced trust based recommender system using ant colony optimization[J]. Applied Intelligence,2016：1-37.
[12] 高升,任思婷,郭军. 基于潜在因子模型的跨领域信息推荐算法[J]. 电信科学,2015,31(7)：75-79.
[13] 张志绮. 基于用户关系的矩阵分解推荐算法研究[D]. 北京：北京交通大学,2016.
[14] 王升升,赵海燕,陈庆奎,等. 基于社交标签和社交信任的概率矩阵分解推荐算法[J]. 小型微型计算机系统,2016,37(5)：921-926.
[15] Song Q,Cheng J,Lu H. Incremental Matrix Factorization via Feature Space Re-learning for Recommender System[C]//The ACM Conference,2015：277-280.
[16] Salakhutdinov R,Mnih A. Probabilistic Matrix Factorization. [J]. Advances in Neural Information Processing Systems,2015：1257-1264.
[17] Hernando A,Bobadilla J,Ortega F. A non—negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model[J]. Knowledge-Based Systems,2016,97(C)：188-202.
[18] Lee H,Kwon J. Improvement of Matrix Factorization-based Recommender Systems Using Similar User Index[J]. International Journal of Software Engineering & Its Applications,2015,9.
[19] Boutet A,Frey D,Guerraoui R,et al. Privacy-preserving distributed collaborative filtering [J]. Computing,2016,98(8)：827-846.
[20] Yu M C,Wu Y C J,Alhalabi W,et al. Research Gate[J]. Computers in Human Behavior,2016,55(PB)：1001-1006.
[21] 朱夏,宋爱波,东方,等. 云计算环境下基于协同过滤的个性化推荐机制[J]. 计算机研究与发展,2014,51(10)：2255-2269.
[22] 杜永萍,黄亮,何明. 融合信任计算的协同过滤推荐方法[J]. 模式识别与人工智能,2014,27(5)：417-425.
[23] 杨欐,钮心忻,黄玮. 基于协同谱聚类的推荐系统托攻击防御算法[J]. 北京邮电大学学报,2015,38(6)：81-86.
[24] 李贵,王爽,李征宇,等. 基于时间加权三部图的分众分类标签推荐算法[J]. 小型微型计算机系统,2016,37(2)：269-274.

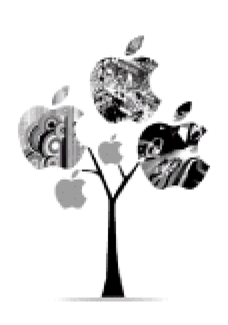
第五篇 原型系统开发

近年来,个性化推荐系统给人们的生活和工作带来了很多便捷。但是随着网络中电影信息的爆炸式增长,简单的模型搜索可能难以满足观影者的需求,同时尽管电影推荐技术越来越得到广泛的应用及普及,但是很多电影系统并没有针对观影者的历史行为进行分析与发掘。另外,在海量数据情况下单机版的算法运行已经不能满足实际的业务需求,同时传统的做法仅仅是存储到数据库,不能满足随时取用的目的。

针对上述问题,为满足用户和电影系统的迫切需要,重视用户的个体需求,本系统运用当下热门的概率矩阵分解技术并结合社交网络中的信任传播机制来改进传统的概率矩阵分解模型;在此基础上,结合Spark集群进行分布式计算、HBase集群构建电影的存储来实现改进的两种概率矩阵分解模型,进一步结合JavaWeb做电影的展示,致力于构建满足不同观影者的偏好各异的界面友好的个性化电影推荐系统。

构建的推荐系统根据观影者的历史行为数据例如点击、购买和收藏等去挖掘用户的偏好信息,进而为其推送感兴趣的电影。另外,构建的推荐系统通过联系观影者和电影一方面帮助观影者发现自己真正想看的电影,另一方面将电影展现在对它感兴趣的观影者面前,从而实现观影者和电影系统的双赢。个性化推荐系统的融入显著提高了观影者的满意度和对电影系统的黏性,进而为电影本身带来了可观的经济效益和社会影响力。

第 14 章



电影推荐原型系统

14.1 引言

本系统用到了目前比较前沿的基于概率矩阵分解和社交网络的推荐技术，与传统的基于内容关键的电影系统有着很大的不同。概率矩阵分解模型相对于传统的基于邻域的模型具有全局的目标函数因而推荐精度较高；同时，由于潜在因子数往往小于系统中的用户数和项目数，算法离线计算的空间复杂度低，这在当今大数据的环境下具有很强的实用价值。

另外，在当今大数据环境下，用户和项目的数量十分巨大，同时用户和项目的数量每天都在持续增加，如每次都重新计算势必会增加计算的任务量，因此搭建了基于 Spark 和 HBase 的集群框架，并基于 Spark 和 HBase 实现两种改进的概率矩阵分解算法，有利于推荐系统在工业界的发展。

14.2 主要功能

如图 14-1 所示是功能架构设计图，该系统主要是在用户查找自己想看的电影时使用，功能包括简介模块、建模一模块、建模二模块、推荐模块、统计分析模块和关于我们模块等内容。

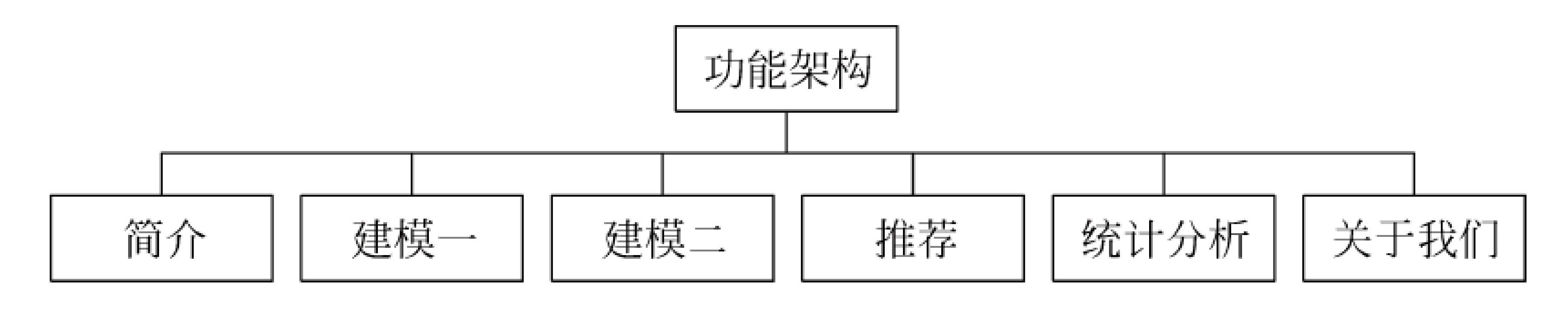


图 14-1　系统功能架构设计图

建模一模块和建模二模块分别对应本系统的两种改进的概率矩阵分解算法，建模之时交互输入算法需要的参数，建模后在推荐模块得到预测的推荐结果，统计分析主要对数据集的内容关键字提取以图表形式直观表达，最后的关于我们模块

是本团队的一些概况。

14.3 关键技术

14.3.1 概率矩阵分解模型

概率矩阵分解是矩阵分解模型中的典型代表，如图 14-2 所示是概率矩阵分解的概率图模型，该模型在 Netflix 等推荐竞赛上大放异彩，吸引了国内外学者的广泛关注。

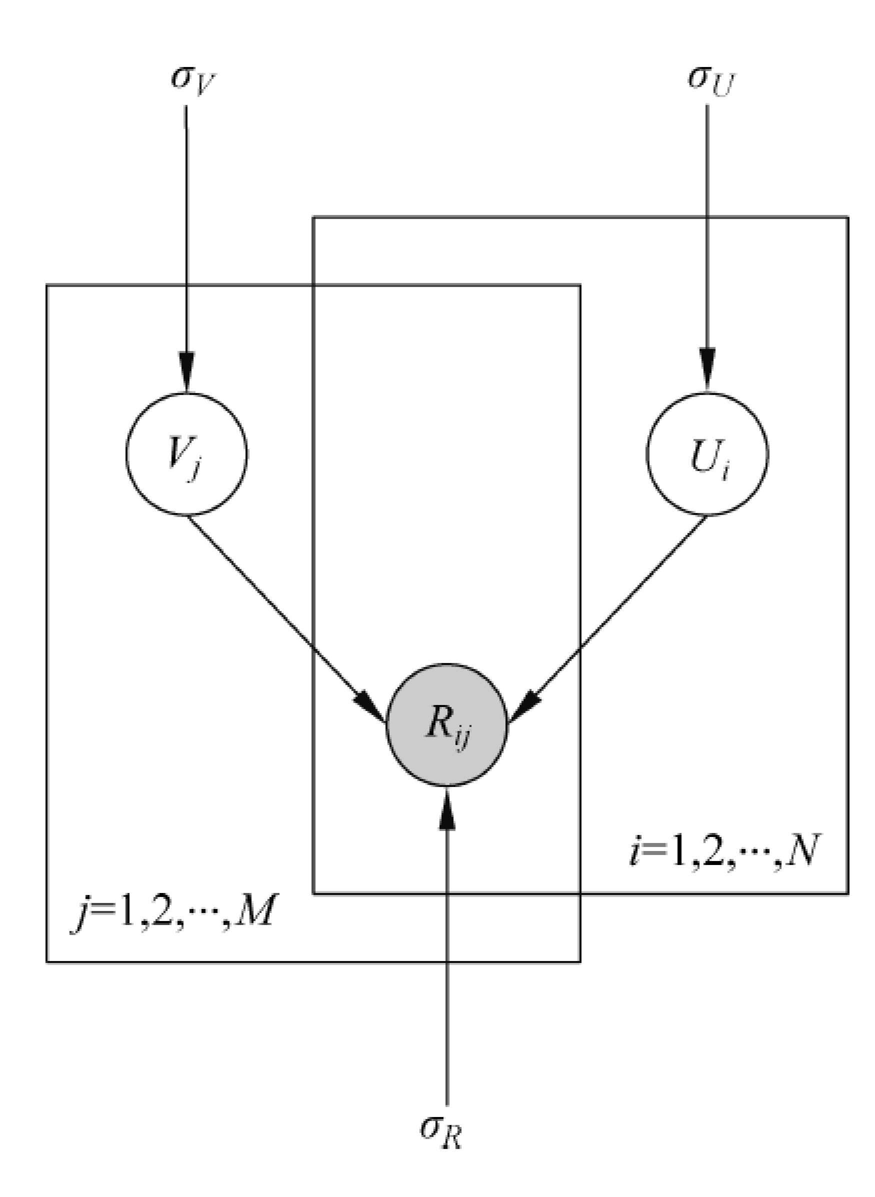


图 14-2 概率矩阵分解的概率图模型

概率矩阵分解的基本思想是在矩阵分解的基础上引入概率的思想，假设用户和商品的特征向量矩阵都符合高斯分布，基于这个假设，用户对商品的喜好程度就是一系列概率的组合问题，利用贝叶斯推导，可得用户和物品的隐式特征的后验概率，结合最大似然估计能得到最终的目标函数。概率矩阵分解模型（与其他矩阵分解算法比较）能较好地适应大规模数据集，时间复杂度随观测数据增长呈线性增长，在数据非常稀疏的情况下有更好的预测性能。

14.3.2 社交网络正则化

出于现实中这样的考虑，一个用户被信任的次数越多，那么越值得被信任；一个用户信任的人越多，那么其对陌生人也就更倾向于信任，但是信任程度会较低。更进一步地说，如果 u_i 信任 u_d，那么这两个用户的用户潜在因子空间也要很相似，相似程度取决于 u_i 对 u_d 的信任程度，通过最小化用户 u_i 和 u_d 的欧氏距离，并将其信任值作为非线性的约束项来约束传统的概率矩阵分解。

如图 14-3 所示为系统推荐原理图，根据用户的潜在需求，首先找到用户的描述文件，在此基础上提取用户的深层次特征，接着结合用户相关数据、项目相关信息以及用户和项目的交互状态形成待推荐的项目列表，最终根据推荐原则获得最优的推荐项目。

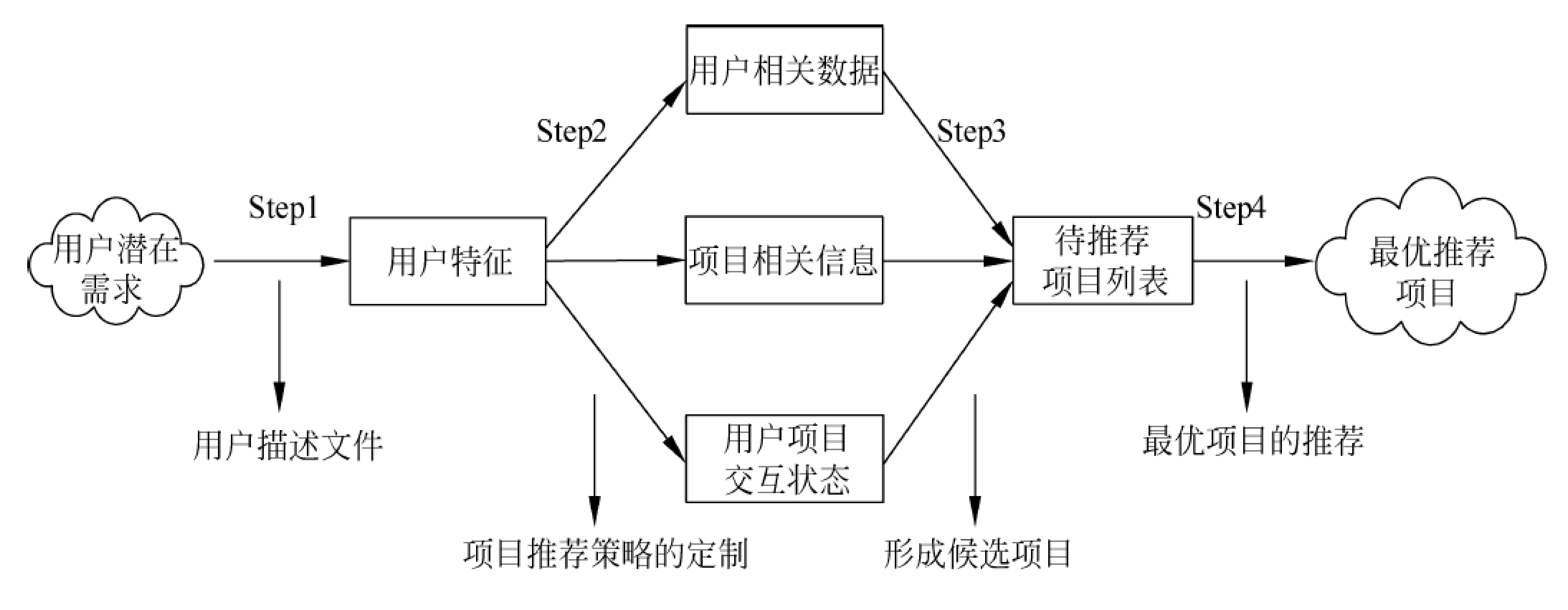


图 14-3 系统推荐原理图

14.4 集群搭建

14.4.1 集群软硬件环境

戴尔笔记本一台，内存 12G，主频 2.7GHz，Windows 10 操作系统，然后安装 VMware Workstation，同时在虚拟机中安装 4 个节点，节点操作系统是 CentOS6，主节点内存是 2G，其他节点的内存是 1G。根据硬件环境规划各个节点的任务，如表 14-1 所列为节点规划，每行标题表示进程，每列标题表示节点名，有对号的表示该节点具有该进程，不填表示没有。

表 14-1 节点规划

	CentOSMaster	StandByNameNode	Slave1	Slave2
NN	√	√		
DN		√	√	√
JNs	√	√	√	
Zk	√	√	√	
ZKFC	√	√		
Spark	√		√	

其中 NN 通过命名空间的镜像文件和编辑日志文件来管理文件系统的命名空间，主要用来接收其他节点的心跳数据来确定各个节点的存活情况，同时也记录着每个文件中各个块所在的数据节点信息。DN 是 Hadoop 文件系统的工作节点，其

根据客户端或者是 NN 的调度来存储和检索数据，并且定期向 NN 发送所存储的块的列表。JNs 的存在是为了同步 ActiveNameNode 和 StandByNameNode 的信息，当 Active 状态的 NN 的命名空间有任何修改时，会通过 JNs 进程通知其他节点，同时 Standby 状态的节点会一直监控 edit log 的变化，进而通过 edit log 把主节点的更改同步到其他节点。Zookeeper 通过原子广播来保证事务的一致性，这样 Client 不论连接到哪个 Server，展示给它都是同一个视图。ZKFC 的 HealthMonitor 主要是监控 NN 主机上的磁盘还是否可用。如表 14-2 所列为集群版本信息。

表 14-2　集群版本信息

软件	版本号
Jdk	1.8.0_19
Zookeeper	3.4.0
Hadoop	2.6.0
Scala	2.11.0
Spark	2.0.2

14.4.2　Spark 集群

Spark 集群是基于内存的/高可靠、高性能的开源分布式并行计算框架，可被用来构建低延迟的大规模数据分析应用程序，同时相对于 Hadoop 集群来说 Spark 集群具有如下优点：运行速度快，简单易用和通用性强。

如图 14-4 所示是搭建好的 Spark 集群。集群主要包括集群地址(URL)、活跃的节点数目(Alive Workers)、集群内存使用情况(Memory in use)和应用状况(Application)等信息。

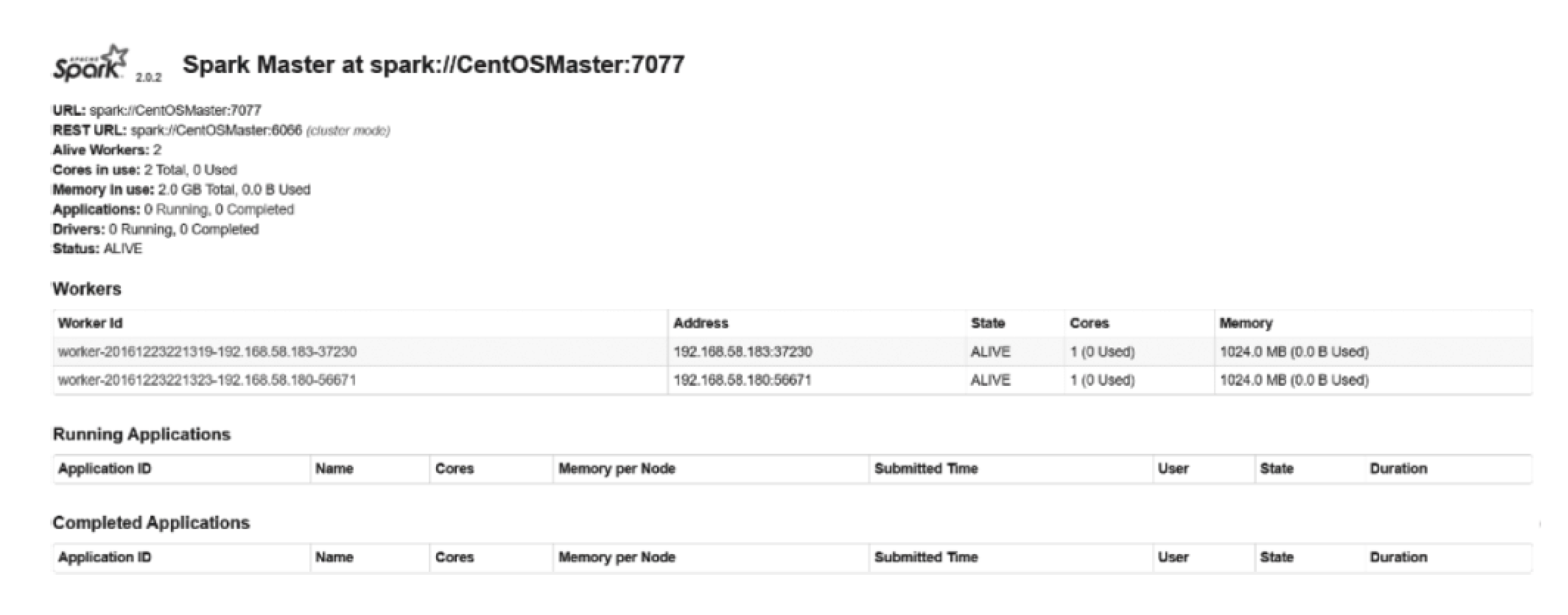


图 14-4　搭建好的 Spark 集群

14.4.3　HBase 集群

HBase 集群是基于 Hadoop 的分布式、可扩展的非关系型大数据存储系统。和 HDFS 不同的是，HBase 可以存储上亿行百万维的数据。另外，集群支持块查

询和布隆过滤器之类的随机查询，方便随机、实时地读取数据，同时易于用 Java 语言进行编写。本系统将推荐结果存储到 HBase 集群，利于以后增量计算方面的研究。

如图 14-5 所示是本章配置好的 HBase 集群，从图中可以看出 HBase 集群的主节点是 CentOSMaster，集群包括三个节点，分别是 Slave1、Slave2 和 StandByNameNode，图中详细显示了各节点服务器名称（ServerName）、集群启动时间（Start time）、版本号（Version）、每秒的请求数量（Requests Per Second）等信息和区域数目（Num Regions）之类的信息，方便用户实时查看集群状态。

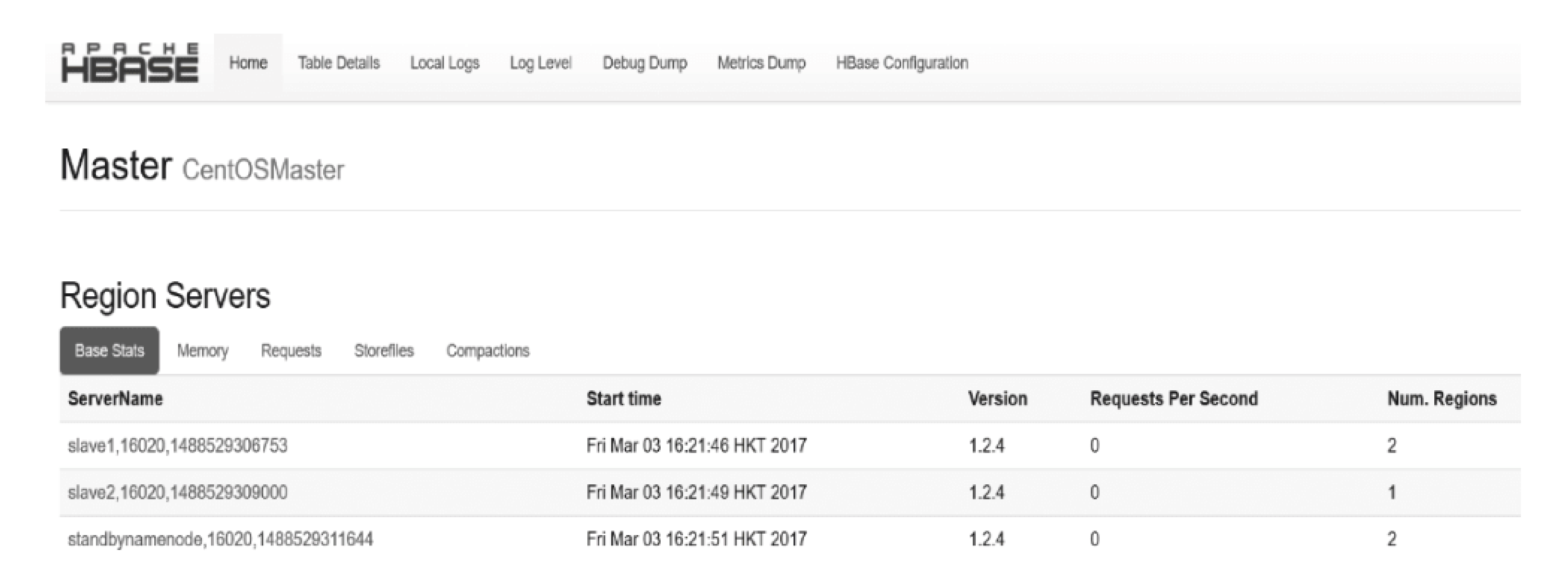


图 14-5 搭建好的 HBase 集群

14.5 系统特点

本系统主要基于 Spark 和 HBase 集群进行开发，主要是利用改进的概率矩阵分解技术为主进行电影推荐，以适应信息过载时代的需求。

具体特点阐述如下：

(1) 提出一种基于交替最小二乘的改进概率矩阵分解算法。首先将用户项目的偏置信息融入到传统的概率矩阵分解算法中；其次为了提升推荐精度，将训练得到的用户项目潜在因子向量作为交替最小二乘的初始值，进而得到用户项目潜在因子矩阵。在此基础上进行预测。

(2) 提出一种社交网络改进的概率矩阵分解算法。首先借鉴用户普遍的认知心理，将信任作为实体决策选择时的一个主观概念，结合客观存在的评分，共同评价用户之间的偏好关系，然后将用户偏置信息和用户间的信任关系融入到传统的概率矩阵分解中，通过随机梯度下降来获取最优解，从而获得对原始用户—项目评分矩阵中缺失的评分值的预测。

(3) 该项目结合主要基于 Spark 集群和 HBase 集群实现，该原型系统为以后的电影系统提供了开发框架，便于系统融合自身的算法模型，同时也为并行化电影推荐系统提供平台支撑，有利于推荐系统在工业界的应用。

14.6 用户使用说明

14.6.1 系统简介界面

如图 14-6 所示是系统简介建模，主要介绍了该系统的功能模块，包括基于 Spark 集群构建矩阵分解模型、在线查询与在线推荐模型、推荐结果存入 HBase 集群以及数据集的统计分析。



图 14-6 系统简介模块

14.6.2 建模一和建模二界面

主要输入数据来源、训练比重、矩阵分解秩、正则系数和训练次数，图 14-7 是建模一模块，图 14-8 是建模二模块，相对于建模一模块主要多了社交系数。

基于Spark MLlib的电影推荐原型系统

简介 建模一 建模二 推荐 统计分析 关于我们

输入或选择参数，提交Spark任务，进行IPMF建模：

输入路径： hdfs://192.168.58.180:8020/MovieLens/ratings.dat

训练比重： 90%

矩阵分解秩： 8

正则系数： 0.1

循环次数： 8

建模

图 14-7 建模一模块

基于Spark MLlib的电影推荐原型系统

简介 建模一 建模二 推荐 统计分析 关于我们

输入或选择参数，提交Spark任务，进行RBPT建模：

输入路径： hdfs://192.168.58.180:8020/MovieLens/ratings.dat

训练比重： 90%

矩阵分解秩： 8

正则系数： 0.065

循环次数： 8

社交系数： 0.1

建模

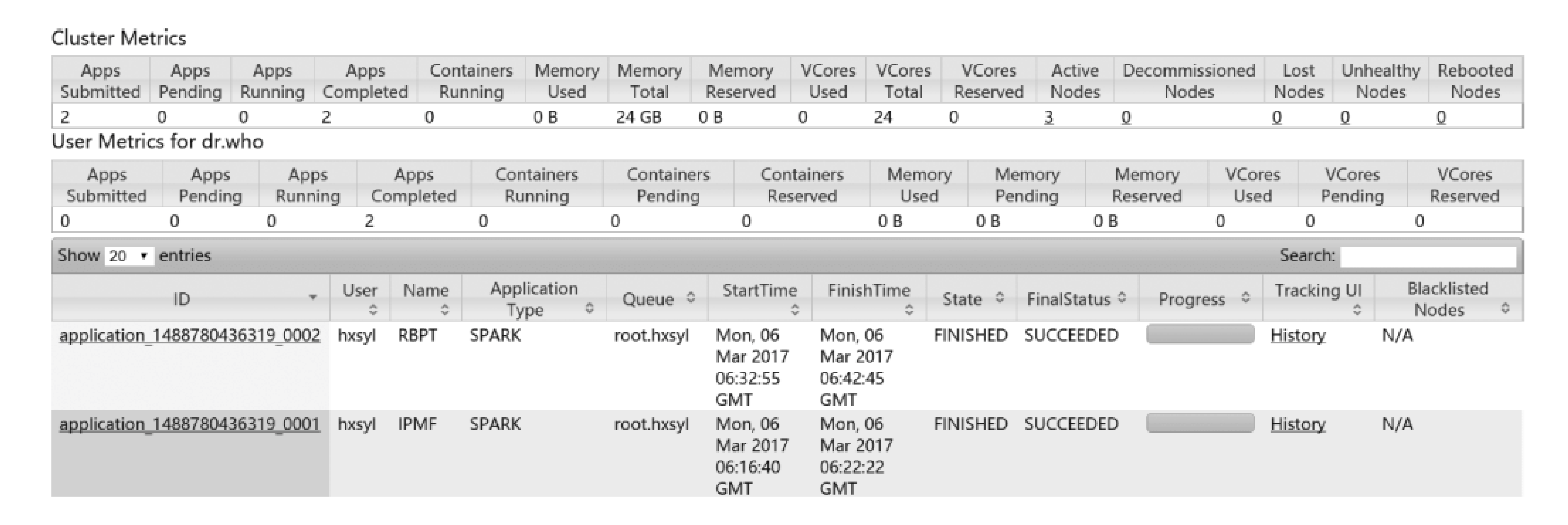
图 14-8 建模二模块

14.6.3 集群界面

如图 14-9 所示为 Spark 运行动态，图中主要包括运行过程中的 Exectors 和 Jobs 的开始时间和结束时间；运行后在 ResourceManager 上显示的状态记录信息。如图 14-10 所示为 ResourceManager 运行记录，图中主要包括运行任务的编号(ID)、算法名(Name)、运行平台(Application Type)和是否运行成功等信息；运行后的推荐结果存储在 HBase 集群中。如图 14-11 所示为运行后的 HBase 集群，用户表中的前两行是电影信息(t_movies)和评分信息(t_ratings)，后两行分别是运行后的推荐结果(t_recommend_by_IPMF 和 t_recommend_by_RBPT)。



图 14-9 Spark 运行动态



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	0	2	0	0 B	24 GB	0 B	0	24	0	3	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCores Reserved
0	0	0	2	0	0	0	0 B	0 B	0 B	0	0	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
application_1488780436319_0002	hxsyl	RBPT	SPARK	root.hxsyl	Mon, 06 Mar 2017 06:32:55 GMT	Mon, 06 Mar 2017 06:42:45 GMT	FINISHED	SUCCEEDED		History	N/A
application_1488780436319_0001	hxsyl	IPMF	SPARK	root.hxsyl	Mon, 06 Mar 2017 06:16:40 GMT	Mon, 06 Mar 2017 06:22:22 GMT	FINISHED	SUCCEEDED		History	N/A

图 14-10　ResourceManager 运行记录

Tables

User Tables　System Tables　Snapshots

4 table(s) in set. [Details]

Namespace	Table Name	Online Regions	Offline Regions	Failed Regions	Split Regions	Other Regions	Description
default	t_movies	1	0	0	0	0	't_movies', {NAME => 'information', VERSIONS => '1'}
default	t_ratings	1	0	0	0	0	't_ratings', {NAME => 'information', VERSIONS => '1'}
default	t_recommend_by_IPMF	1	0	0	0	0	't_recommend_by_IPMF', {NAME => 'information', VERSIONS => '1'}
default	t_recommend_by_RBPT	1	0	0	0	0	't_recommend_by_RBPT', {NAME => 'information', VERSIONS => '1'}

图 14-11　运行后的 HBase 集群

14.6.4　看过的电影界面

如图 14-12 所示是用户看过的电影信息图，首先输入用户 ID，然后点击“查询”，在下方显示用户评分过的电影信息条目数，在其下方以列表形式显示用户看过的电影信息，主要内容包括电影编号、电影名称、电影所属的类别标签和评分信息。



MovieId	电影名	标签	评分
1	Toy Story (1995)	Animation|Children's|Comedy	5.0
1022	Cinderella (1950)	Animation|Children's|Musical	5.0
1028	Mary Poppins (1964)	Children's|Comedy|Musical	5.0
1029	Dumbo (1941)	Animation|Children's|Musical	5.0
1035	Sound of Music, The (1965)	Musical	5.0
1097	E.T. the Extra-Terrestrial (1982)	Children's|Drama|Fantasy|Sci-Fi	4.0
1193	One Flew Over the Cuckoo's Nest (1975)	Drama	5.0
1197	Princess Bride, The (1987)	Action|Adventure|Comedy|Romance	3.0

图 14-12　用户看过的电影信息图

14.6.5　推荐电影界面

选择 IPMF 推荐(对应建模一)得到如图 14-13 所示的建模一 IPMF 模型推荐结果,选择 RBPT 推荐(对应建模二)得到如图 14-14 所示的建模二 RBPT 模型推荐结果,最终推荐结果按推荐分按递减次序排列。



图 14-13　建模一 IPMF 推荐结果



图 14-14　建模二 RBPT 推荐结果

14.6.6　统计分析界面

如图 14-15 所示是统计分析界面,从图中能得到不同年龄段的观影者所看过的电影数目信息,以便于后期研究。

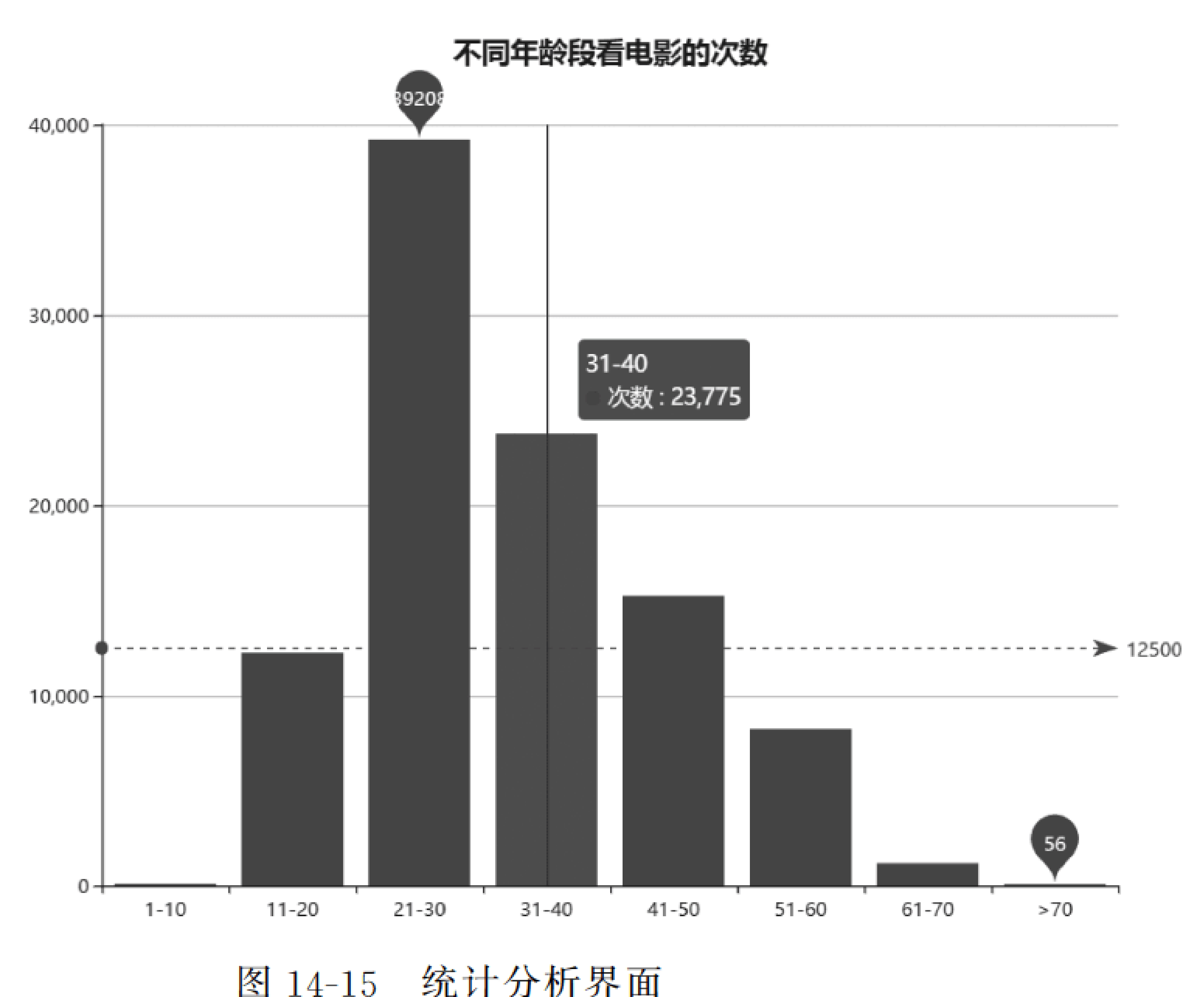


图 14-15 统计分析界面

参考文献

[1] 杨志伟. 基于 Spark 平台推荐系统研究[D]. 合肥：中国科学技术大学，2015.

[2] 胡于响. 基于 Spark 的推荐系统的设计与实现[D]. 杭州：浙江大学，2015.

[3] 郑凤飞. 基于 Spark 的并行推荐算法的研究与实现[D]. 成都：西南交通大学，2016.

[4] 李文栋. 基于 Spark 的大数据挖掘技术的研究与实现[D]. 济南：山东大学，2015.

[5] 尹绪森. SparkMLlib：矩阵参数的模式[J]. 程序员，2014(8)：108-112.

[6] 张明敏. 基于 Spark 平台的协同过滤推荐算法的研究与实现[D]. 南京：南京理工大学，2015.

[7] Zeng J，Leng B，Xiong Z. 3-D object retrieval using topic model[J]. Multimedia Tools and Applications，2015，74(18)：7859-7881.

[8] Koren Y. Factorization meets the neighborhood：a multifaceted collaborative filtering model [C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM，2008：426-434.

[9] Lian D，Zhao C，Xie X，et al. GeoMF：joint geographical modeling and matrix factorization for point-of-interest recommendation[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM，2014：831-840.

[10] Gillis N，Vavasis S A. Fast and Robust Recursive Algorithmsfor Separable Nonnegative Matrix Factorization[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence，2014，36(4)：698-714.

[11] Sun D L，Fevotte C. Alternating direction method of multipliers for non-negative matrix

factorization with the beta-divergence[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2014: 6201-6205.

[12] Hastie T, Mazumder R, Lee J D, et al. Matrix completion and low-rank SVD via fast alternating least squares[J]. Journal of Machine Learning Research, 2015, 16(1): 3367-3402.

[13] Espig M, Hackbusch W, Khachatryan A. On the Convergence of Alternating Least Squares Optimisation in Tensor Format Representations[J]. Mathematics, 2015, 72(22): 4303.

[14] Tichavský P, Phan A H, Cichocki A. Partitioned Alternating Least Squares Technique for Canonical Polyadic Tensor Decomposition[J]. IEEE Signal Processing Letters, 2016, 23(7): 993-997.

[15] Bauer S, Stefan J, León F P. Hyperspectral image unmixing involving spatial information by extending the alternating least-squares algorithm[J]. Technisches Messen, 2015, 82(4): 174-186.

[16] Guo G, Zhang J, Sun Z, et al. Librec: A java library for recommender systems[C]//Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization. 2015: 955-963.

[17] 张吉沣. 基于 Hadoop 的推荐算法研究[D]. 北京: 北京工业大学, 2015.

[18] 梅忠, 肖如良, 张桂刚. 基于受约束偏置的概率矩阵分解算法[J]. 计算机系统应用, 2016, 25(5): 113-117.

[19] 陆园丽. 基于非负矩阵分解的鲁棒推荐算法研究[D]. 秦皇岛: 燕山大学, 2015.

[20] Ortega F, Hernando A, Bobadilla J, et al. Recommending items to group of users using Matrix Factorization based Collaborative Filtering[J]. Information Sciences, 2016, 345(C): 313-324.

[21] Zhao X, Niu Z, Chen W, et al. A hybrid approach of topic model and matrix factorization based on two-step recommendation framework[J]. Journal of Intelligent Information Systems, 2015, 44(3): 335-353.

[22] Zhao Y, Li S, Hou J. Link Quality Prediction via a Neighborhood-Based Nonnegative Matrix Factorization Model for Wireless Sensor Networks[J]. International Journal of Distributed Sensor Networks, 2015, 2015(1): 1-8.

[23] Solovyev S A, Tordeux S. An efficient truncated SVD of large matrices based on the low-rank approximation for inverse geophysical problems[J]. Université De Pau Et Des Pays De Ladour, 2015: 592-609.

[24] Mori K, Nguyen T, Harada T, et al. An Improvement of Matrix Factorization with Bound Constraints for Recommender Systems[C]//Iiai International Congress on Advanced Applied Informatics, 2016: 103-106.

[25] Aleksandrova M, Brun A, Boyer A, et al. Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem[J]. Journal of Intelligent Information Systems, 2016: 1-33.

[26] Hanhua Chen, Hai Jin, Xiaolong Cui. 微博系统中一种混合关注对象推荐方法[J]. Science China Information Sciences, 2017, 60(1): 012102.

[27] Ma T, Zhou J, Tang M, et al. Social Network and Tag Sources Based Augmenting

Collaborative Recommender System[J]. Ieice Transactions on Information & Systems, 2015, E98. D(4): 902-910.

[28] Hong M, Jung J J. MyMovieHistory: Social Recommender System by Discovering Social Affinities Among Users[J]. Cybernetics & Systems, 2016, 47(1-2): 88-110.

[29] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]//International Conference on Machine Learning. ACM, 2008: 880-887.

[30] Rendle S. Factorization Machines with libFM[J]. Acm Transactions on Intelligent Systems & Technology, 2012, 3(3): 57.